



TUGAS AKHIR - TE145561

**PERANCANGAN KOMUNIKASI DARI SISTEM
MONITORING PADA AUTOMATION SORTING LINE
SYSTEM**

Ivan Dharmawan Setianto
NRP 10311500000007

Pembimbing
Eko Setijadi, ST., MT., Ph.D.
Imam Arifin, ST., MT.
Mohammad Hafid, S.Pd

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman ini sengaja dikosongkan



FINAL PROJECT - TE145561

***COMMUNICATION DESIGN OF MONITORING SYSTEM
ON AUTOMATION SORTING LINE SYSTEM***

Ivan Dharmawan Setianto
NRP 10311500000007

Supervisor
Eko Setijadi, ST., MT., Ph.D.
Imam Arifin, ST., MT.
Mohammad Hafid, S.Pd.

*Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2018*

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

**“PERANCANGAN KOMUNIKASI DARI SISTEM
MONITORING PADA AUTOMATION SORTING LINE SYSTEM”**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 10 Juli 2018



Ivan Dharmawan Setianto
NRP. 10311500000007

Halaman ini sengaja dikosongkan

**PERANCANGAN KOMUNIKASI DARI SISTEM
MONITORING PADA AUTOMATION SORTING LINE
SYSTEM**

TUGAS AKHIR

**Diajukan-Guna Memenuhi Sebagaimana Persyaratan Untuk
Memperoleh Gelar Ahli Madya Pada**

**Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Pembimbing I,

**Menyetujui,
Pembimbing II,**

Pembimbing III

Eko Setiadi, ST, MT, Ph.D

Amam Arifin, ST, MT,

M Hafid, S.Pd

NIP. 196602181991021061

NIP. 197302222002121001

**SURABAYA
JUNI, 2018**

Halaman ini sengaja dikosongkan

PERANCANGAN KOMUNIKASI DARI SISTEM *MONITORING* PADA *AUTOMATION SORTING LINE SYSTEM*

Ivan Dharmawan Setianto
10311500000007

Pembimbing I : Eko Setijadi, ST., M.T., Ph.D
Pembimbing II : Imam Arifin, ST., M.T.
Pembimbing III : Mohammad Hafid, S.Pd

ABSTRAK

Komunikasi sangat dibutuhkan dalam suatu sistem yang ada di industri sebagai pengiriman atau penerimaan data/informasi dari kegiatan yang dilakukan oleh suatu mesin dalam suatu pabrik. Dengan adanya komunikasi operator juga dapat mengetahui banyaknya pengambilan data per detik dari suatu mesin. Selain itu, melalui jaringan komunikasi kegiatan mesin di pabrik yang menghubungkan beberapa perangkat seperti sensor, kontroler, layar tatap muka, serta *Personal Computer* yang bekerja dapat memberikan hasil dan dapat dilihat oleh operator dari *control room*. Kontroler sendiri memiliki kegunaan untuk mengatur jalannya sistem secara otomatis. Selain kontroler yang digunakan sebagai pengendali, ada layar tatap muka yang digunakan untuk berinteraksi dengan mesin dengan sebutan *Human Machine Interface* (HMI) berupa layar tatap muka dan dapat berfungsi mempermudah operator berinteraksi terhadap mesin. Dengan adanya alat ini, operator dapat mengetahui bagian eror, PLC hanya menerima sinyal eror yang dikirim dari mesin, lalu dari kemudian dilanjut ke HMI untuk menampilkan bagian yang eror pada mesin tersebut. Beberapa perangkat tersebut perlu dihubungkan dengan ethernet agar proses pengiriman dan penerimaan data atau informasi dapat berjalan tepat waktu, sehingga tidak menghambat jalannya proses dari mesin. Cara kerjanya adalah dengan menyamakan alamat IP dari semua perangkat. Kemudian untuk memicu *barcode reader* agar dapat mengambil data *barcode* dari box yang lewat secara otomatis yaitu dengan membuat program PLC yang ditujukan ke alamat IP dari *barcode reader*.

Kata Kunci : Komunikasi, kontroler, layar tatap muka.

Halaman ini sengaja dikosongkan

COMMUNICATION DESIGN OF MONITORING SYSTEM ON AUTOMATION SORTING LINE SYSTEM

Ivan Dharmawan Setianto
10311500000007

Supervisor I	: Eko Setijadi, ST.,MT.,Ph.D.
Supervisor II	: Imam Arifin,ST., MT.
Supervisor III	: Mohammad Hafid, S.Pd

ABSTRACT

Communication is needed in the form of an existing system in the industry as the transmission or receipt of data / information from the activities undertaken by the factory machinery. With the communication operator can also know which data per second from the machine. In addition, through communication networks the activity of machines connected to devices such as sensors, controllers, screens, and personal computers that work can provide results and views by operators of the control room. The controller has the utility to set the system automatically. In addition, there are also screens used to connect with machines with the Human Machine Interface (HMI) and can also be used by other operators such as machines. With this tool, the operator can know the error part, PLC only receive messages sent from the machine, from then proceeded to HMI to see the error on the machine. Some devices are required with ethernet in order that the process of sending and receiving data or information can run on time, not blocking the process of the machine. It works by equating the IP address of all devices. Then to access the barcode reader in order to retrieve the barcode data from the box that runs automatically by creating a PLC program that redirects to the IP address of the barcode reader.

Keywords: *Communication, Controller, Interface.*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul: **"Perancangan Komunikasi dari Sistem Monitoring pada Automation Sorting Line System."**

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk memperoleh gelar diploma pada Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada berbagai pihak yang telah memberikan bantuan dan dukungan dalam penyelesaian Tugas Akhir, terutama kepada bapak, ibu, dan keluarga yang senantiasa memberikan semangat dan doa yang tidak terbatas. Kemudian bapak Eko Setijadi, ST., MT., Ph.D. dan bapak Imam Arifin, ST, MT., serta bapak Mohammad Hafid, S.Pd selaku pembimbing yang telah banyak memberikan bimbingan, saran, serta masukan yang sangat berarti bagi penulis. Lalu untuk seluruh dosen Departemen Teknik Elektro Otomasi yang telah banyak memberikan ilmu selama penulis menempuh perkuliahan. Tak lupa teman-teman di Departemen Teknik Elektro Otomasi ITS yang tidak dapat penulis sebutkan satu persatu, terutama rekan-rekan anggota Laboratorium Sistem Komputer dan Otomasi BB202 yang selalu memberikan bantuan pemikiran serta semangat dalam menyelesaikan Tugas Akhir ini. Serta semua pihak yang tidak bisa penulis sebutkan. Besar harapan penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Departemen Teknik Elektro Otomasi khususnya.

Surabaya, 10 Juli 2018



Ivan Dharmawan Setianto

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR ...Error! Bookmark not defined.	
LEMBAR PENGESAHAN	ix
ABSTRAK.....	xi
ABSTRACT.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL.....	xxi
1 BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Metodologi.....	4
1.6 Sistematika Penulisan.....	5
1.7 Ruang Lingkup.....	6
2 BAB 2 KOMUNIKASI PADA AUTOMATION	
<i>SORTING LINE SYSTEM.....</i>	7
2.1 Konveyor	7
2.2 <i>Automation Sorting Line System.....</i>	9
2.3 <i>Cognex DM60 Barcode Reade</i>	10
2.4 <i>Dataman Software</i>	11
2.5 Ethernet.....	12
2.6 <i>Programmable Logic Controller (PLC).....</i>	12
2.6.1 CPU (<i>Central Processing Unit</i>).....	14
2.6.2 Memori	15
2.6.3 <i>Input dan output Terminal Block.....</i>	15
2.6.4 <i>CPU Unit Operations Status Indicator</i>	16
2.7 Sysmac Studio	16
2.7.1 Variabel pada Sysmac Studio	17
2.7.2 POU (<i>Program Organizing Unit</i>)	19
2.7.3 Membuat Tipe Data.....	19
2.8 HMI (<i>Human Machine Interface</i>)	19

2.9	Omron NB7W-TW01B.....	20
2.10	NB Designer	25
2.11	Hub	25
2.12	Relay	26
2.13	Motor DC	27
2.14	Motor Servo	28
2.15	<i>Limit Switch</i>	30
2.16	Sensor Inframerah.....	30
3	BAB 3 PERENCANAAN SISTEM.....	33
3.1	Ruang Lingkup Kerja.....	33
3.2	Diagram Alir Keseluruhan	34
3.3	Perancangan Perangkat Keras	35
3.4	Perancangan Komunikasi	36
3.5	Perancangan Perangkat Lunak	37
3.5.1	<i>Setup</i> konfigurasi HMI dengan PLC dengan NB Designer.....	37
3.5.2	Perancangan HMI	38
3.5.3	<i>Setting</i> Parameter <i>Barcode Reader</i> ke PC Melalui <i>DataMan Software</i>	41
3.5.4	Membuat Modul <i>Input Output Assembly</i> Untuk Menghubungkan PLC dengan Sensor Cognex DM60	42
3.5.5	Function Block Dataman.....	47
3.5.6	Konfigurasi Sysmac Studio pada PC dengan Omron NX1P2	48
4	BAB 4 PENGUJIAN DAN ANALISIS.....	49
4.1	Pengujian Human Machine Interface Omron NB7W-TW01B	49
4.2	Pengujian Pembacaan <i>input text</i> HMI dan <i>push button</i> pada HMI.....	50
4.3	Pengujian pembacaan <i>Barcode</i> di <i>software</i> Sysmac Studio.....	52
4.4	Pengujian menampilkan pesan <i>error</i> di HMI.....	56
4.5	Pengujian proses pengiriman data	57
4.6	Pengujian proses waktu pemicuan.....	58
5	BAB 5 PENUTUP.....	59
5.1	Kesimpulan.....	59
5.2	Saran.....	59
	DAFTAR PUSTAKA	61
	LAMPIRAN	63
	RIWAYAT PENULIS.....	119

DAFTAR GAMBAR

Gambar 2.1 Konveyor	8
Gambar 2.2 <i>Automation Sorting Line System</i>	9
Gambar 2.3 Cognex DM 60 Barcode Reader	10
Gambar 2.4 Tampilan utama <i>Dataman Software</i>	11
Gambar 2.5 PLC Omron NX1P2-9024DT	14
Gambar 2.6 <i>Block Terminal Input</i>	15
Gambar 2.7 Lampu indikator pada CPU NX1P2.....	16
Gambar 2.8 <i>Software Sysmac Studio</i> dari Omron	17
Gambar 2.9 Dimensi Omron NB7W-TW01B	20
Gambar 2.10 Port COM1 pada NB7W-TW01B	21
Gambar 2.11 Port USB pada Omron NB7W-TW01B.....	22
Gambar 2.12 Port USB SLAVE pada Omron NB7W-TW01B	23
Gambar 2.13 Port Ethernet.....	24
Gambar 2.14 Hub.....	25
Gambar 2.15 Relay.....	26
Gambar 2.16 Motor DC.....	27
Gambar 2.17 Motor Servo	28
Gambar 2.18 Skema Pulsa Kendali Motor Servo	29
Gambar 2.19 <i>Limit Switch</i>	30
Gambar 2.20 Sensor Inframerah	31
Gambar 3.1 Ruang Lingkup Kerja	33
Gambar 3.2 <i>Flowchart</i> Sistem Keseluruhan.....	34
Gambar 3.3 Konfigurasi Keseluruhan Perangkat Keras	35
Gambar 3.4 Konfigurasi Perangkat pada Jaringan Ethernet.....	36
Gambar 3.5 Menghubungkan HMI dengan PLC melalui NB Designer.	37
Gambar 3.6 Menyamakan IP HMI dan PLC melalui NB Designer... 38	
Gambar 3.7 Halaman Utama pada Sistem <i>Monitoring</i>	38
Gambar 3.8 Halaman <i>Error</i> pada HMI	39
Gambar 3.9 Halaman <i>Counter Box</i> pada HMI	40
Gambar 3.10 Tampilan menu Quick Setup pada <i>Dataman Software</i> 41	
Gambar 3.11 <i>Timing chart Acquisition Sequence</i>	43
Gambar 3.12 <i>Timing chart Result Sequence</i>	44
Gambar 3.13 <i>Flowchart</i> proses pengambilan data oleh <i>barcode reader</i>	45
Gambar 3.14 Pembacaan data pada <i>Dataman</i> oleh PLC.....	46

Gambar 3.15 Alur Pemicuan *Barcode reader* dengan PLC.....47

Gambar 3.16 *Function block dataman*.....47

Gambar 3.17 Menghubungkan Sysmac Studio dengan ethernet IP...48

Gambar 4.1 HMI pada *Automation Sorting Line System*.....49

Gambar 4.2 HMI ketika belum terhubung dengan PLC.....50

Gambar 4.3 Data *barcode* 9788679912077 yang berhasil dibaca Oleh PLC.....55

Gambar 4.4 Terjadinya *error* pada *sorting line* A dikarenakan penuh dengan box..... 56

Gambar 4.8 Grafik waktu pemicuan58

DAFTAR TABEL

Tabel 2.1	Tipe variable dengan penjelasannya pada Sysmac Studio..	18
Tabel 2.2	Penjelasan mengenai kaki <i>port</i> COM1 HMI.....	21
Tabel 2.3	Penjelasan mengenai kaki <i>port</i> COM2 pada HMI	22
Tabel 2.4	Tabel penjelasan kaki pada USB <i>Host</i>	23
Tabel 2.5	Fungsi dari kaki-kaki pada pin USB SLAVE	23
Tabel 2.6	Fungsi dari kaki-kaki pin ethernet.....	24
Tabel 3.1	Modul <i>input assembly</i>	42
Tabel 3.2	Modul <i>output assembly</i>	42
Tabel 4.1	Hasil pengujian komunikasi pada <i>line A</i>	51
Tabel 4.2	Hasil pengujian komunikasi pada <i>line B</i>	51
Tabel 4.3	Hasil pengujian komunikasi pada <i>line C</i>	52
Tabel 4.4	Pengujian penekanan tombol pada HMI.....	52
Tabel 4.5	Hasil baca dari <i>barcode reader</i> yang dikirim ke PLC dengan barcode 9778679912077	53
Tabel 4.6	Hasil baca dari <i>barcode reader</i> yang dikirim ke PLC dengan <i>barcode</i> 0123456789012.....	53
Tabel 4.7	Hasil baca dari <i>barcode reader</i> yang dikirim ke PLC dengan <i>barcode</i> 8911609119892.....	54
Tabel 4.8	Hasil baca dari <i>barcode reader</i> yang dikirim ke PLC dengan <i>barcode</i> 9781855682979.....	55
Tabel 4.9	Pengujian Status pada HMI.....	56

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pelaksanaan magang selama 6 bulan bertempat di PT. Industrial Robotic Automation. Perusahaan ini merupakan perusahaan yang memiliki fokus pada otomasi dalam kegiatan perindustrian untuk meningkatkan efisiensi sumber daya manusia, kualitas produk, fleksibilitas, penelitian dan inovasi teknologi. Perusahaan ini mengembangkan sebuah proyek yang sangat dibutuhkan oleh industri seperti sistem *Robotic Box palletizer* yang berfungsi untuk proses peletakan box pada pallet sebelum diletakkan di rak gudang untuk dikirim ke konsumen. Dalam sistem ini terdapat subsistem seperti *sorting line conveyor*, *weight box checker*, *automatic shuttle pallet*, *automatic pallet dispenser*, serta *pallet wrapper*.

Pada saat magang selama 3 bulan pertama, kami diberi tugas untuk mengawasi proyek *Automatic Box Palletizer* di PT. Bentoel Internasional Investama selama 24 jam. Jam kerja tersebut dibagi menjadi 3 shift dimana shift 1 dimulai pukul 08.00-16.00, shift 2 dimulai pada pukul 16.00-24.00, dan shift 3 yang dimulai pukul 24.00-08.00. Jam kerja tersebut selalu berganti agar setiap orang merasakan waktu bekerja pada ketiga shift tersebut. Setelah 3 bulan selesai, dilanjutkan untuk persiapan Tugas Akhir membuat salah satu sub sistem dari *Robotic Box Palletizer*. memprogram sensor pendeteksi *barcode* dan *patern* tetapi dengan proyek ini dan dikerjakan 3 orang terlalu mudah untuk dijadikan topik tugas akhir mahasiswa. Jadi kami dan pembimbing berunding untuk memilih satu alat yang ada pada proyek tersebut untuk dijadikan topik tugas akhir kami bertiga.

Pada Tugas Akhir kali disarankan untuk memilih *Automation Sorting Line System*, yang terdiri dari konveyor yang digerakan oleh motor, pendorong yang digunakan untuk proses pemilahan serta *barcode reader* untuk mengambil data dari *barcode* pada box yang akan dipilah. Konveyor merupakan suatu sistem mekanik yang memiliki fungsi memindahkan barang dari suatu tempat ke tempat lain. Konveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan[1]. Seiring berjalannya waktu, konveyor tidak hanya digunakan untuk memindahkan barang dari suatu

tempat, tetapi dapat digunakan juga untuk memilah beberapa produk yang telah dihasilkan oleh suatu industri. Konveyor pemilah ini disebut dengan *Automation Sorting Conveyor* yang berfungsi untuk memilah produk secara otomatis dan berkelanjutan. Pada alat ini terdapat berbagai macam, aktuator, sensor, dll yang digunakan untuk menjalankan mesin tersebut. Pada sistem ini kontroler, *interface*, serta terdapat sensor yang harus berkomunikasi dengan jaringan ethernet wilayah setempat (LAN). Komunikasi data merupakan suatu proses pengiriman dan penerimaan data/informasi dari dua atau lebih perangkat seperti laptop, komputer, printer dan alat komunikasi lain yang terhubung dalam sebuah jaringan. Baik lokal maupun yang luas, seperti internet. Pada dasarnya komunikasi data merupakan proses pengiriman informasi di antara dua titik menggunakan kode biner melewati saluran transmisi dan peralatan *switching*, bisa antara komputer dan komputer, komputer dengan terminal, atau komputer dengan peralatan, atau peralatan dengan peralatan.

Komunikasi juga dibutuhkan oleh suatu sistem yang terdapat di industri seperti perangkat sensor sampai dengan *personal computer* sebagai pengiriman atau penerimaan data/informasi dari kegiatan yang dilakukan oleh suatu mesin dalam suatu pabrik untuk melakukan pengawasan terhadap jalannya pendistribusian produk. Selain itu, melalui jaringan komunikasi kegiatan mesin di pabrik yang menghubungkan beberapa perangkat seperti sensor, kontroler, layar tatap muka, serta *personal computer* yang bekerja dapat memberikan hasil yang dapat dilihat oleh operator dari *control room*. Dengan adanya komunikasi operator juga dapat mengetahui banyaknya pengambilan data per detik dari suatu mesin.

Salah satunya adalah sistem *monitoring* yang dapat digunakan untuk mengawasi kegiatan yang ada di industri. Sistem *monitoring* sendiri sangat diperlukan agar mempermudah pengawasan dari suatu proyek yang sedang berjalan. Dengan adanya sistem ini, kita tidak perlu kesusahan untuk mencari eror pada suatu bagian mesin. Jika terjadi eror, cukup melihat dari layar *interface* operator sudah dapat mengetahui posisi dari eror pada bagian mesin tersebut. Sinyal eror tersebut muncul dari *plant* yang mengalami masalah lalu mengirimkannya ke kontroler, dan kontroler sendiri mengirim sinyal ke HMI untuk ditampilkan. Kontroler sendiri memiliki kegunaan untuk mengatur jalannya sistem secara otomatis. Beberapa alat tersebut perlu dihubungkan dengan ethernet agar proses pengiriman dan penerimaan data atau informasi

dapat berjalan tepat waktu, sehingga tidak menghambat jalannya proses dari mesin.

Pada Tugas Akhir ini, dibuat Sistem *monitoring* untuk mengawasi dan mengatur jalannya *Automation Sorting Line System*. Pada HMI yang dibuat, dapat memasukan kode *barcode* untuk menyalurkan box ke *sorting line* yang diinginkan.

1.2 Perumusan Masalah

Perumusan masalah yang akan dibahas pada tugas akhir ini meliputi komunikasi antara *barcode reader*, PLC, serta HMI yang bekerja mengirim data satu sama lain, serta menampilkan performa pembacaan dari *barcode reader* dalam membaca *barcode* pada box yang lewat pada *personal computer* (PC), jika masukan dari HMI sesuai dengan pembacaan sensor *barcode* akankah menggerakkan servo untuk mengarahkan box ke *sorting line* yang dituju atau tidak menggerakkan aktuator agar box tersebut menuju ke *conveyor reject*.

1.3 Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah mengatasi salah satu permasalahan yang ada di industri dengan membuat konveyor pemilah yang sangat dibutuhkan untuk memilah produk yang dihasilkan suatu pabrik. Pada system tersebut terdapat beberapa perangkat yang memiliki fungsi berbeda agar dapat bekerja dalam proses mengirim dan menerima data satu sama lain. Agar memudahkan operator maka dibuat desain *interface* untuk sistem *monitoring* yang dapat digunakan untuk memasukan digit *barcode* agar masukan tersebut dapat terbaca oleh PLC. Kemudian PLC juga menerima masukan dari hasil yang telah didapat oleh sensor untuk menyamakan dengan masukan dari HMI agar dapat menggerakkan aktuator untuk diarahkan ke *sorting line* yang dituju. Selain itu dari sistem *monitoring* sendiri dapat menampilkan eror pada bagian tertentu.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Menampilkan pesan dari eror yang muncul pada mesin
2. *Barcode* yang digunakan merupakan *barcode* yang sudah dipatenkan pada program
3. Menampilkan jumlah box yang masuk ke *sorting line*

4. Menggunakan *power supply* 24V.
5. PLC yang digunakan yaitu Omron NX1P2-9024DT.
6. Barcode reader yang digunakan adalah Cognex DM60.
7. HMI yang digunakan yaitu Omron NB7W-TW01B.
8. *Power Supply* yang digunakan memiliki tegangan 24VDC.

1.5 Metodologi

Metodologi yang digunakan pada pengerjaan tugas akhir ini terdiri dari empat tahap. Tahap pertama adalah melakukan studi literatur. Pada tahap studi literatur, dilakukan pengumpulan dasar teori yang menunjang penguasaan terhadap permasalahan yang berhubungan dengan perancangan komunikasi dari sistem *monitoring* pada *automation sorting line system*. Studi literatur berupa rujukan dalam bentuk jurnal internasional maupun nasional, buku, dan artikel di internet, maupun dari tugas akhir mahasiswa sebelumnya.

Hal yang dipelajari dalam tahap pertama untuk komunikasi dan perangkat lunak. Untuk komunikasi, hal-hal yang dipelajari antara lain menghubungkan beberapa perangkat menggunakan jaringan ethernet, dan mengatur alamat IP dari setiap perangkat agar saling terhubung. Sedangkan untuk perangkat lunak, hal-hal yang dipelajari antara lain adalah pemrograman PLC berbasis *structure text*, penggunaan *software* NB Designer untuk membuat tampilan *interface*, serta penggunaan *software* DataMan untuk menampilkan hasil pembacaan teks pada box.

Pada tahap kedua yaitu tahap perancangan desain HMI yang akan digunakan sebagai layar untuk bertatap muka dengan mesin yang digunakan, membuat program PLC untuk memicu *barcode reader* agar dapat membaca *barcode* yang lewat. Menentukan parameter yang digunakan untuk jarak pembacaan sensor *barcode* pada DataMan *software*.

Tahap selanjutnya adalah mengkonfigurasikan ketiga perangkat tersebut agar saling terhubung dan saling memberikan informasi agar dapat melanjutkan ke proses selanjutnya. Dengan menyamakan alamat IP pada ketiga perangkat tersebut, komunikasi akan terhubung dan proses *monitoring* dapat berjalan.

Tahapan terakhir merupakan penyusunan laporan. Dalam penyusunan laporan diperlukan beberapa hal, pengumpulan seluruh analisa data dari semua percobaan, serta penyusunan referensi.

1.6 Sistematika Penulisan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi Tugas Akhir yang dibuat.

BAB II TEORI DASAR

Menjelaskan teori yang dapat dijadikan landasan dan pendukung dalam perencanaan dan pembuatan alat yang dibuat

BAB III PERANCANGAN SISTEM

Membahas perencanaan dan pembuatan tentang hardware yang meliputi desain mekanik dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

BAB IV PENGUJIAN DAN ANALISIS SISTEM

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

BAB V PENUTUP

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

Halaman ini sengaja dikosongkan

BAB II

KOMUNIKASI PADA *AUTOMATION SORTING LINE SYSTEM*

Sistem *Monitoring* merupakan suatu sistem yang sangat dibutuhkan suatu kegiatan yang ada di industri. Latar belakang dari penelitian tentang perancangan komunikasi dari sistem *monitoring* pada *Automation Sorting Line System* berawal dari menghubungkan beberapa perangkat yang memiliki peran yang berbeda dan saling bekerja sama untuk mendapat dan memberikan informasi atau data dalam pengawasan dari pemilahan pada konveyor pemilah dari beberapa produk yang dihasilkan oleh ruang produksi. Pemilahan tersebut berdasarkan angka *barcode* yang dimasukkan ke HMI yang kemudian dikirim ke PLC. Lalu *barcode reader* yang berfungsi untuk membaca *barcode* dan mengirim data ke PLC, kemudian di PLC sendiri membandingkan data yang dimasukan dari HMI dengan data yang telah dibaca oleh *barcode reader* untuk menggerakkan aktuator agar produk tersebut dapat diarahkan ke *sorting line* yang dituju.

2.1 Konveyor

Sistem konveyor memiliki sejarah panjang. Sistem ini telah menjadi bagian dari penanganan material selama lebih dari satu abad. Pengangkutan bahan curah oleh ban konveyor atau berjalan mulai sekitar tahun 1795, tetapi penemuan ini pada dasarnya hanya digunakan untuk mengangkut biji-bijian pada jarak yang sangat pendek. Sistem sabuk konveyor yang paling awal dibuat dari kulit, kanvas, atau sabuk karet yang berjalan di atas tempat tidur kayu yang rata. Dengan ini, para insinyur memutuskan bahwa konveyor merupakan cara ekonomis dan efisien untuk mengangkut bahan curah dalam jumlah besar dari satu lokasi ke lokasi lain. Perkembangan cepat sistem ini terdapat pada abad ke 20. Konveyor digunakan di area Minneapolis, Saint Paul untuk membongkar *shingles* dari gerbong kereta. Tidak lama setelah itu, konveyor mulai membuat penampilan ditempat lain seperti konveyor sabuk yang digunakan untuk memindahkan barang di pabrik[1].

Pada tahun 1908, Hymle Goddard dari Logan *Company* merupakan orang pertama yang mematenkan *roller conveyor*. Sekitar tahun 1919 produksi otomotif mulai menggunakan konveyor, dan di era

1920an, konveyor mulai populer untuk menangani barang-barang yang di produksi massal dalam pabrik. Selama perang dunia ke-2, produsen membuat bahan sintetis untuk membuat konveyor karena kurangnya komponen alami. Sekarang ini, sabuk konveyor dibuat dari polimer dan bahan sintetis yang tidak ada habisnya. Konveyor sudah digunakan hampir disemua industri.

Ada juga beberapa kemajuan teknologi pada konveyor, yang termasuk pemeliharaan dan kontrol sistem. Pemeliharaan konveyor sangat sulit karena sistemnya secara permanen dan dihubungkan dan diperbaiki. Perubahan produksi dan perawatan standar biasanya membutuhkan *downtime* lama dan biaya yang cukup besar. Seringkali, konveyor harus diganti dengan sistem yang lebih baru sebelum rusak. Selama tahun 70-an sampai dengan 90-an, insinyur peralatan konveyor mengembangkan dan menyempurnakan *roller* konveyor bertenaga internal dengan pulley bermotor agar dapat menghilangkan kebutuhan pemeliharaan yang mahal. Dengan kemajuan baru dan inovasi teknologi, sistem konveyor tetap canggih, bersama dengan sistem penanganan material otomatis lainnya. Komputer sekarang dapat mengontrol dan peningkatan otomatisasi membantu sistem menjadi lebih efisien[2].



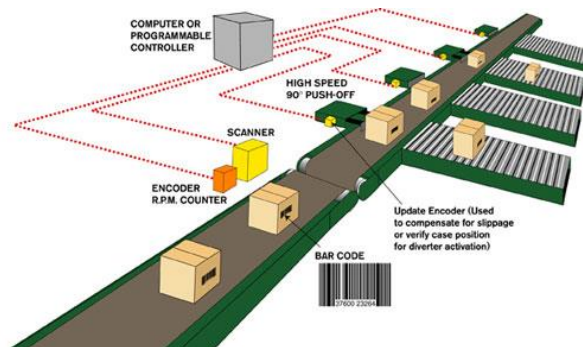
Gambar 2.1 Konveyor

Seiring perkembangan zaman, konveyor tidak hanya digunakan untuk mengantarkan barang saja, tetapi dapat digunakan untuk memilah barang sesuai dengan ketentuan yang ada di industri pemakai. Selain memindahkan barang dari satu tempat ke tempat lain, konveyor pemilah juga dapat digunakan untuk memilah beberapa produk berbeda yang

dihasilkan di suatu industri. Pemilahan tersebut dapat ditentukan oleh ukuran, jenis, berat, serta kode barang yang diangkut oleh konveyor. Pada sistem konveyor sekarang ini juga terdapat PC yang digunakan untuk mengawasi produk yang dipilah oleh alat tersebut. Selain PC, terdapat kontroler terprogram yang digunakan untuk mengatur jalannya sistem secara otomatis dan berkelanjutan, serta layar tatap muka sebagai visualisasi dari sistem tersebut agar mempermudah manusia untuk menjalankan sistem ini.

2.2 Automation Sorting Line System

Pada era globalisasi sekarang ini otomatisasi sudah banyak diaplikasikan di industri. Otomatisasi dalam sistem penyortiran meningkatkan akurasi, menghemat biaya dan waktu tenaga kerja contohnya seperti sistem pemilahan produk di suatu industri[3]. Sistem pemilahan yang dilakukan melalui konveyor ini sangat diperlukan oleh industri. Sistem ini dapat digunakan untuk proses pemilahan berbagai macam produk yang dihasilkan oleh suatu industri. Untuk pemilahannya, kebanyakan ditentukan dari *barcode* yang tercantum pada box yang dilewatkan pada konveyor. Pada sistem ini terdapat kontroler yang digunakan untuk mengatur jalannya system agar dapat bergerak secara otomatis dan berkelanjutan sesuai dengan kebutuhan yang ada di industri. Selain itu terdapat, HMI (*Human Machine Interface*) yang berfungsi untuk memudahkan operator dalam mengoperasikan *Automation Sorting Line System* ini. Berikut merupakan gambar dari sistem terseBut yang ditunjukan pada Gambar 2.0.



Gambar 2.2 *Automation Sorting Line System*

Selain itu dibutuhkan beberapa sensor yang berfungsi pada jalannya pemilahan tersebut seperti *barcode reader* yang digunakan untuk mengambil data dari box yang lewat dan juga terdapat sensor limit switch dan photoelectric yang digunakan untuk mendeteksi jika box yang lewat akan terus terdeteksi sampai berhasil dipilah. Untuk actuator yang digunakan untuk memilah sendiri dapat menggunakan motor servo maupun *pusher pneumatic* untuk mendorong barang yang lewat agar masuk ke *sorting line* yang dituju. Dengan adanya alat ini dapat mengurangi tenaga manusia dalam hal pemilahan beberapa produk yang dihasilkan oleh suatu industri. Pada alat ini juga terdapat HMI yang digunakan untuk mengatur jalannya sistem dan digunakan juga untuk menampilkan info seperti jumlah box yang berhasil disortir dan munculnya pesan error dari sistem tersebut[3].

2.3 Cognex DM60 Barcode Reader

Barcode reader yang digunakan bermerek Cognex tipe DM60 (DataMan 60) yang diproduksi oleh Cognex Corporation yang memiliki fitur seperti *monitoring* dengan performa tinggi dengan tetap terpasang untuk pembacaan kode 1D dan 2D. Pada sensor ini terdapat kabel yang terhubung langsung dengan konektor DB15 yang menyediakan *power*, I/O, dan juga konektivitas RS 232. Selain itu juga tersedia *port ethernet* yang berfungsi sebagai komunikasi dari DM60[4]. Bentuk dari barcode reader yang digunakan ditunjukkan pada Gambar 2.3.



Gambar 2.3 Cognex DM60 barcode reader.

DataMan 60 (*barcode reader*) sendiri memiliki beberapa jenis mode pemacu untuk mengambil gambar dari barcode seperti berikut:

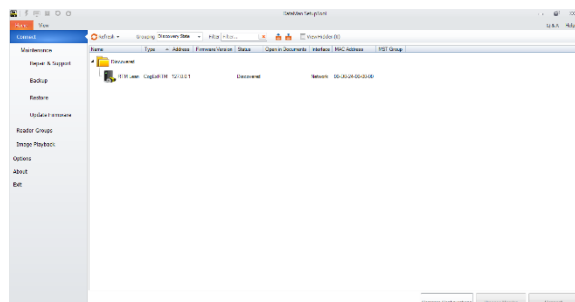
1. *Self* : Pada interval yang dikonfigurasi, pembaca secara otomatis mendeteksi dan mengode ulang kode dalam bidang pandangnya. Jika mengatur *delay re-read* yang lebih tinggi

daripada interval pemicu, ada *output* kode hanya sekali sampai kode keluar dari bidang tampilan untuk durasi penundaan pembacaan ulang.

2. *Single* (pemicu eksternal): Mengakuisisi satu gambar dan mencoba untuk menguraikan simbol apa pun yang dikandungnya, atau lebih dari satu simbol dalam kasus di mana *multicode* diaktifkan. Pembaca bergantung pada sumber pemicu eksternal.
3. *Presentation*: Memindai, menerjemahkan kode dan melaporkan kode tunggal di bidang tampilan. Pembaca bergantung pada internal mekanisme waktu untuk mendapatkan gambar.
4. *Manual*: Mulai mengambil gambar saat pengguna menekan tombol pemicu pada *barcode reader*, dan terus memperoleh gambar sampai simbol ditemukan dan diterjemahkan atau dilepaskan tombol.

2.4 Dataman Software

Software yang digunakan untuk mengatur parameter dari *barcode reader* adalah Dataman. Ketika mengambil gambar, data tersebut muncul pada menu *quick setup* dan kemudian dikirim ke perangkat lain. Untuk mengatur parameter dari Cognex DM60, alamat IP dari PC harus disamakan dengan *reader* yang digunakan. Dataman *software* ini juga dapat digunakan untuk melihat hasil dari pembacaan *barcode* yang telah dibaca oleh *barcode reader* dan menampilkan hasil dari *barcode* yang menunjukkan terbacanya *barcode* yang dibaca atau tidak[4]. Berikut merupakan tampilan dari *Dataman software* pada gambar 2.4.



Gambar 2.4 Tampilan awal dari *Dataman Software*.

2.5 Ethernet

Jaringan ethernet merupakan teknologi jaringan komputer berdasarkan pada kerangka jaringan area lokal (LAN). Jaringan LAN merupakan suatu media transmisi bersama serta rangkaian *hardware* dan *software* untuk membuat perangkat saling terhubung. Sistem komunikasi melalui Ethernet membagi aliran data ke dalam paket individual yang disebut *frame*. Setiap *frame*, berisi alamat sumber dan tujuan serta pengecekan data eror sehingga data yang rusak dapat dideteksi dan dikirim kembali. Ethernet merupakan protokol LAN yang memungkinkan setiap PC untuk mengakses *network*[5]. Jenis-Jenis Ethernet Dilihat dari kecepatannya, Ethernet terbagi menjadi empat jenis, antara lain:

- 10 Mbit/detik, kecepatan dari Ethernet (standar yang digunakan: 10Base2, 10Base5, 10BaseT, 10BaseF)
- 100 Mbit/detik, yang sering disebut sebagai *Fast Ethernet* (standar yang digunakan: 100BaseFX, 100BaseT, 100BaseT4, 100BaseTX)
- 1000 Mbit/detik atau 1 Gbit/detik, merupakan Gigabit Ethernet (standar yang digunakan: 1000BaseCX, 1000BaseLX, 1000BaseSX, 1000BaseT).

2.6 Programmable Logic Controller (PLC)

PLC merupakan perangkat elektronik yang beroperasi secara digital dan didesain untuk pemakaian di lingkungan industri[4], dimana sistem ini menggunakan memori yang dapat diprogram untuk penyimpanan secara internal dengan instruksi-instruksi yang mengimplementasikan fungsi spesifik seperti logika, urutan, perwaktuan, pencacahan dan operasi aritmatik untuk mengontrol mesin atau proses melalui modul I/O digital maupun analog. PLC dirancang untuk menggantikan suatu rangkaian relay sekuensial dalam suatu sistem kontrol[6]. Selain dapat diprogram, alat ini juga dapat dikendalikan, dan dioperasikan oleh orang yang tidak memiliki pengetahuan di bidang pengoperasian komputer secara khusus.

PLC ini memiliki bahasa pemrograman yang mudah dipahami dan dapat dioperasikan bila program yang telah dibuat dengan menggunakan *software* yang sesuai dengan jenis PLC yang digunakan sudah diunggah. Alat ini bekerja berdasarkan *input* yang ada dan tergantung dari keadaan pada suatu waktu tertentu yang kemudian akan menyalakan atau mematikan *output*. Logika 1 menunjukkan bahwa

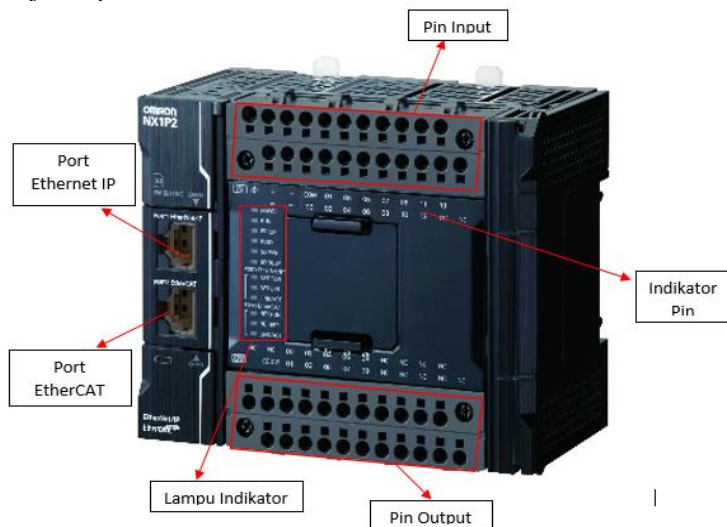
keadaan yang diharapkan terpenuhi sedangkan logika 0 berarti keadaan yang tidak terpenuhi. Istilah logika digunakan karena pemrogramannya berkaitan dengan penerapan logika dan operasi *switching*. Perangkat *input* seperti *switch*, dan perangkat *output* seperti motor, dikendalikan terhubung ke PLC dan kemudian kontroler memonitor *input* dan *output* sesuai dengan mesin atau proses.

Prinsip kerja sebuah PLC adalah menerima sinyal masukan dari proses yang dikendalikan lalu melakukan serangkaian instruksi logika terhadap sinyal masukan tersebut sesuai dengan program yang tersimpan dalam memori lalu menghasilkan sinyal keluaran untuk mengendalikan aktuator atau peralatan lainnya[7]. Dalam industri-industri yang ada sekarang ini, kehadiran PLC sangat dibutuhkan terutama untuk menggantikan sistem wiring atau pengkabelan yang sebelumnya masih digunakan dalam mengendalikan suatu sistem. Dengan adanya alat ini pengguna dapat mengendalikan beberapa mesin yang memiliki fungsi berbeda dengan PLC. Selain itu jika salah satu bagian dari system akan diubah, maka pengubahannya hanya dilakukan pada program yang terdapat pada computer dalam waktu yang relatif singkat kemudian program tersebut diunggah ke PLC. Jika tidak menggunakan alat ini, misalnya seperti menggunakan relay maka perubahannya dilakukan dengan cara mengubah pengkabelan yang membutuhkan waktu yang lebih lama. Dengan alat ini pengkabelan yang dilakukan lebih sederhana dibandingkan menggunakan relay.

Salah satunya adalah kontroler Sysmac NX-series yang diproduksi oleh Omron yang merupakan kontroler otomasi mesin generasi baru yang menyediakan fungsionalitas dan kinerja berkecepatan tinggi yang diperlukan untuk kontrol alat berat. Alat menyediakan keamanan, keandalan, dan pemeliharaan yang diperlukan dari pengendali industri. Pengontrol NX-series menyediakan fungsionalitas PLC Omron sebelumnya, dan mereka juga menyediakan fungsionalitas yang diperlukan untuk kontrol gerakan. Terutama, NX-series NX1P2 kontroler yang dapat melakukan operasi I / O dengan NX Unit atau *Option Board* terlampir, dan dengan I / O internal.

Omron menawarkan perangkat kontrol Sysmac Series baru yang dirancang dengan spesifikasi komunikasi terpadu dan spesifikasi antarmuka pengguna. Pengontrol otomatis seri NX ini merupakan bagian dari Sysmac Series. Anda dapat menggunakannya bersama dengan EtherCAT, produk Sysmac lainnya, dan Perangkat lunak Sysmac Studio untuk mencapai fungsi dan kemudahan pengoperasian

yang optimal. Dengan sistem yang dibuat dari produk Sysmac, pengguna dapat menghubungkan komponen dan menugaskan sistem melalui konsep dan kegunaan terpadu. Bentuk dari PLC Omron NX1P2 ditunjukkan pada Gambar 2.5.



Gambar 2.5. PLC Omron NX1P2-9024DT

Pada PLC ini memiliki port EtherCAT untuk menghemat ruang. Dengan adanya *port* ini dapat mempermudah saat membangun sistem dengan berbagai Unit NX[11]. PLC ini memiliki *subsistem* keselamatan pada EtherCAT. Pengguna dapat menggunakan unit kontrol keamanan NX-series untuk mengintegrasikan kontrol keselamatan secara berurutan dan gerak sistem kontrol sebagai *subsistem* pada EtherCAT.

2.6.1 CPU (Central Processing Unit)

CPU memiliki fungsi untuk mengendalikan pengoperasian keseluruhan sistem pengontrol yang dapat diprogram. Modul prosesor ini biasanya terletak di salah satu sisi rakitan rak. Pada CPU ini berisi integrasi chip sirkuit yang mencakup satu dan lebih mikroprosesor, memori chip, dan sirkuit yang memungkinkan data disimpan dan diambil dari memori [7]. Prosesor ini terdiri dari tiga bagian utama: pusat unit pemrosesan (CPU), unit logika aritmatika (ALU) dan ingatan. Unit pengolah pusat adalah otak dari PLC. Prinsip fungsi CPU adalah

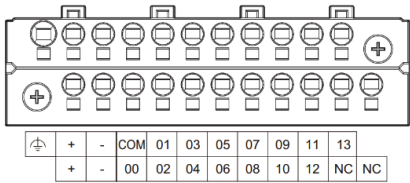
menafsirkan dan mengeksekusi berbasis komputer program yang disimpan secara permanen di memori prosesor. Program-program ini ditulis oleh pabrikan PLC untuk mengaktifkan memori yang tidak stabil, yang berarti bahwa jika daya masukan ke ini chip dihapus, isinya akan hilang.

2.6.2 Memori

Memori merupakan tempat penyimpanan data sementara dan tempat menyimpan program yang harus dijalankan, dimana program tersebut merupakan hasil terjemahan dari *ladder diagram* yang dibuat oleh pengguna. Sistem memori pada PLC juga mengarah pada teknologi *flash memory*. Dengan menggunakan *flash memory* akan sangat mudah bagi pengguna untuk melakukan pemrograman secara berulang-ulang. Selain itu pada *flash memory* juga terdapat EPROM yang dapat dihapus berulang-ulang. Sistem memori dibagi dalam beberapa blok dimana setiap blok memiliki fungsi tersendiri. Beberapa bagian dari memori digunakan untuk menyimpan status dari *input* dan *output*, sementara bagian memori yang lain digunakan untuk menyimpan variabel yang digunakan pada program seperti nilai *timer* dan *counter*. PLC memiliki suatu rutin kompleks yang digunakan untuk memastikan memori PLC tidak rusak. Hal ini dapat dilihat lewat lampu indikator pada PLC.

2.6.3 Input dan Output Terminal Block

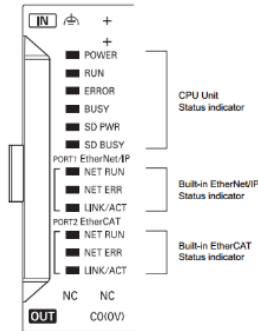
Pada terminal *input* dan *output* dari PLC ini terdapat terminal *ground*, COM, +/- . Masing-masing dari terminal tersebut memiliki fungsi dan kegunaannya tersendiri. Seperti terminal +/- yang berfungsi untuk mendapatkan tegangan dari catu daya. Kemudian ada terminal *ground* yang berfungsi sebagai pengaman ketika terjadi kebocoran arus, dan *port* COM yang memiliki kegunaan untuk tempat masuknya *test lead* kutub, serta 14 *terminal input* dan 10 *terminal output* pada PLC ini. Berikut bentuk dari *terminal input* dari PLC NX1P yang ditunjukkan pada Gambar 2.6.



Gambar 2.6 Block terminal input.

2.6.4 CPU Unit Operations Status Indicator

Pada CPU NX1P2 terdapat *CPU Unit Status Indicator*, *Built-in Ethernet/IP Status Indicator*, serta *Built-in EtherCAT Status Indicator* yang menunjukkan pada operasi yang sedang berlangsung pada PLC[8],[9]. Lampu indikator pada CPU Omron NX1P2 ditunjukkan pada Gambar 2.7.



Gambar 2.7 Lampu indikator pada Omron NX1P2.

Power indicator akan menyala berwarna hijau jika PLC aktif dan jika indikator tidak aktif menyatakan bahwa PLC tidak aktif. Pada *RUN indicator*, jika lampu menyala berwarna hijau menandakan bahwa unit CPU dalam operasi normal pada mode RUN. Kemudian ketika lampu RUN aktif dengan berkedip dengan interval 2 detik kemudian berkedip dengan interval selama 0,5 detik menunjukkan bahwa sedang terjadinya proses pengunggahan program ke PLC, tetapi jika lampu indicator menyala dengan interval 2 detik selama 30 detik atau lebih menandakan terjadinya eror dalam proses pengunggahan. Jika lampu sudah mati menunjukkan bahwa program telah berhasil diunggah ke PLC. Kemudian jika lampu indicator eror menyala berwarna merah yang menunjukkan adanya banyak kesalahan pada PLC, sedangkan jika berkedip masih terdapat eror tetapi proses masih dapat berlanjut.

2.7 Sysmac Studio

Software keluaran Omron seperti Sysmac Studio ini menyediakan lingkungan pengembangan terintegrasi untuk mengatur, memprogram, dan *controller* Sysmac NX-series dan pengontrol otomasi mesin lainnya, serta *slave* EtherCAT.

Sysmac Studio menyediakan lingkungan untuk pemrograman dengan variabel. Tidak perlu khawatir tentang alamat memori. Hal ini menghilangkan kebutuhan untuk menunggu definisi alamat memori untuk perangkat keras sebelum dimulainya pengembangan perangkat lunak. Perangkat keras dan perangkat lunak dapat dirancang secara independen dan dikembangkan secara paralel. POU (*Program Organization Unit*) yang mencakup program, *function*, dan *function block* dapat digunakan untuk merancang pemrograman yang tidak bergantung pada satu sistem tertentu[9]. Dengan ini meningkatkan penggunaan untuk pemrograman. Tampilan awal Sysmac Studio ditampilkan pada Gambar 2.8.



Gambar 2.8 Sysmac Studio Automation Software

Sysmac Studio menyediakan lingkungan untuk pemrograman dengan variabel dan POU. Pemrograman dirancang dengan POU (program, *function*, dan *function block*). Program-program tersebut kemudian ditugaskan untuk mengurutan eksekusi program yang telah didefinisikan. Hal ini mengurangi saling ketergantungan program, oleh karena itu memungkinkan lebih dari satu pemrogram untuk bekerja dengan mudah pada saat yang bersamaan. Tugas dari variabel ke perangkat keras dan definisi hubungan antara informasi yang dibagi antara program yang berbeda dapat diatur kapan saja.

2.7.1 Variabel Registration pada Sysmac Studio

Variabel Registration merupakan variabel untuk menyimpan data yang digunakan dalam algoritma kontrol di Unit CPU NX-series. Pengguna dapat menentukan jenis variabel berikut.

Tabel 2.1 Tipe variable dengan penjelasannya pada Sysmac Studio.

Tipe Variabel	Deskripsi
<i>Global Variables</i>	Pengguna dapat membaca dan menulis variabel global dari setiap POU (program, fungsi, atau blok fungsi). Dengan menyatakan variabel global dalam tabel variabel global. Pengguna dapat menggunakan variabel tersebut pada POU yang berbeda, sehingga dapat membaca dan menulis variabel global dari setiap POU yang dijalankan pada Unit CPU di mana variabel global dideklarasikan. Namun, POU yang berjalan pada Unit CPU lain tidak dapat mengakses variabel ini. Variabel perangkat yang dihasilkan secara otomatis dari konfigurasi Unit dan konfigurasi <i>slave</i> secara otomatis terdaftar sebagai variabel global.
<i>Local Variables</i>	Variabel lokal hanya dapat digunakan dalam satu POU. Variabel lokal dideklarasikan dalam tabel variabel lokal untuk POU. Anda tidak dapat mengakses nilai variabel lokal dari luar POU. Bahkan jika pengguna mendeklarasikan variabel lokal dengan nama yang sama dalam POU yang berbeda, variabel-variabel tersebut tidak berbagi ruang memori yang sama.
<i>Function Block Instance Variables</i>	<i>Function Block Instance Variables</i> adalah nama yang ditetapkan untuk <i>Function Block Instance</i> sehingga pengguna dapat mengelola secara individual. <i>Function Block</i> yang telah memiliki nama dapat mempertahankan variabel internalnya.
<i>Derivative Variables (Structure Unions, and Enumerations)</i>	Variabel turunan memiliki tipe data yang ditentukan pengguna.

Tipe Variabel	Deskripsi
	<i>Derivative variable</i> tersebut mengelompokkan variabel-variabel tipe data yang berbeda ke dalam satu variabel. Pengguna dapat memberi nama apa pun ke tipe data. pengguna harus menentukan nama variabel yang menggunakan tipe data yang ditentukan dan setidaknya satu anggota dalam variabel itu.

2.7.2 POU (*Program Organization Unit*)

POU adalah unit yang digunakan untuk membangun sebuah program. POU dapat berupa program yang berisikan *function*, *function block*, *ladder diagram* serta *structure text*. Pengguna harus memulai dengan terlebih dahulu mendaftarkan *ladder diagram*, *structure text*, *function*, serta *function block* di bawah pemrograman POU di *Multiview Explorer Sysmac Studio*, dan kemudian mengkodekan POU. Dengan adanya POU ini dapat mempermudah pengguna untuk mencari nama-nama dari program, *function* atau *function block* yang telah dibuat, dimana komponen-komponen tersebut memiliki fungsi yang berbeda, tetapi memiliki tujuan yang sama untuk menggerakkan suatu sistem[9].

2.7.3 Mendaftarkan Tipe Data

Membuat tipe data melibatkan mendefinisikan tipe data baru dengan menggabungkan tipe data yang ada. Jenis data yang terdaftar menggunakan editor tipe data dan dapat dipilih dari daftar tipe data pada *Editor Variabel* yang digunakan seperti jenis data lainnya. Pengguna dapat menentukan tipe data yang merupakan struktur, perserikatan, atau enumerasi[9].

2.8 HMI (*Human Machine Interface*)

Human Machine Interface merupakan perangkat antarmuka berbasis komputer berupa tampilan penghubung antara manusia dengan mesin atau peralatan yang dikendalikan[9]. Perangkat ini dapat membuat visualisasi dari teknologi atau sistem secara nyata, visualisasi tersebut dilengkapi dengan data – data yang nyata dan sesuai dengan keadaan di lapangan. Selanjutnya visulisasi tersebut ditampilkan pada monitor – monitor diruang kendali secara *realtime* bahkan sudah dapat diakses secara *online* melalui peralatan elektronik dimanapun dan

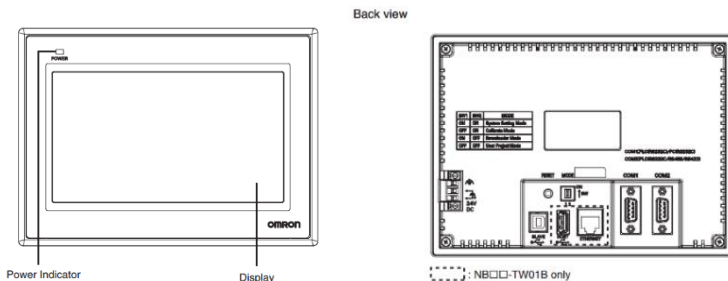
kepun selama ada jaringan internet. Untuk proses skala kecil seperti di *sub system* maka HMI yang digunakan dapat berupa tampilan *touchscreen* yang lebih sederhana[10].

HMI dapat digunakan untuk mengakses sistem setiap saat untuk berbagai tujuan, seperti menampilkan jumlah produk, menampilkan kesalahan pada mesin, menampilkan status pada mesin, dimana operator melakukan pengendalian pada mesin. Penggunaan alat ini memiliki beberapa keuntungan, misalnya penggunaan kode warna, bentuk dan lain sebagainya, sehingga mudah untuk dikenali. Pada Tugas Akhir ini HMI harus bekerja secara terintegrasi dengan PLC. PLC sendiri mengambil informasi data dari *barcode reader*, dan data tersebut disamakan dengan masukan dari HMI untuk melakukan langkah selanjutnya seperti menggerakkan servo untuk mendorong box agar masuk ke *sorting line* yang dituju.

2.9 OMRON NB7W-TW01B

NB7W-TW01B merupakan HMI (*Human Machine Interface*) berupa layar sentuh (*Touchscreen*) yang memiliki performa tinggi. Lampu latar LED yang digunakan lebih ramah lingkungan, hemat energi dan memiliki masa pakai lebih lama dibandingkan lampu latar CCFL tradisional. Perangkat display seri NB (PT: Programmable Terminal) dapat digunakan untuk display informasi dan operasi masukan. HMI ini dapat menunjukkan status operasi sistem dan perangkat secara grafis.

Pembuatan program pada tampilan Omron NB7W-TW01B disusun berdasarkan pengalamatan program yang terdapat pada PLC. Komunikasi hubungan Omron NB7W-TW01B dengan Omron NX1P2 menggunakan ethernet yang tersambung pada hub[11]. Dimensi dari HMI Omron yang digunakan ditunjukkan pada Gambar 2.9.



Gambar 2.9 Dimensi Omron NB7W-TW01B.

Pada HMI Omron NB7W-TW01B terdapat *serial port* COM1, *port* COM2, *USB Host*, *USB Slave*, dan *ethernet interface* yang masing-masing memiliki fungsi tersendiri seperti berikut:

1. *Serial port* COM1

Port ini merupakan soket tipe D dengan 9-pin. *Port* ini mendukung fungsi komunikasi RS-232C, membuatnya dapat dihubungkan ke pengontrol yang memiliki fungsi RS-232C, dan ini juga dapat digunakan untuk mengunduh program atau *debugging* untuk produk. Untuk *port* serial COM 1 ditunjukkan pada Gambar 2.10.



Gambar 2.10 *Port* COM1 pada NB7W-TW01B.

Setiap pin dari serial *port* COM1 memiliki fungsi dan kegunaan masing-masing. Fungsi dari setiap pin pada *port* COM1 didefinisikan pada Tabel 2.2 sebagai berikut.

Tabel 2.2 Penjelasan mengenai kaki dari *port* COM1 HMI.

Pin	Simbol	I/O	Fungsi
1	NC	-	-
2	SD	O	Mengirim Data*
3	RD	I	Menerima Data*
4	RS	O	Meminta untuk mengirim
5	CS	I	Hapus untuk dikirim
6	DC+5V	-	DC+5V <i>output</i> (Arus maksimal 250 mA)
7	NC	-	-
8	NC	-	-
9	SG	-	<i>Signal Ground</i>

*Pin 4 dan 5 tidak digunakan, sehingga tidak sesuai dengan fungsi RS atau CS

2. Port COM2

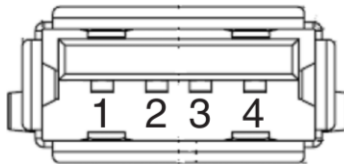
Port serial COM2 merupakan *port* soket tipe D dan memiliki 9-pin. *Port* ini mendukung fungsi komunikasi RS-232C / RS-485 / RS-422A. Gambar dari pin pada *port* COM2 mirip dengan *port* COM1. Fungsi dari kaki pin tersebut akan dijelaskan pada Table 2.3 di bawah ini.

Table 2.3 Penjelasan mengenai kaki *port* COM2 pada HMI.

Pin	Simbol	I/O	Fungsi		
			RS-232C	RS-285	RS-422A
1	SDB+	I/O	-	-	Mengirim data(+)
2	SD	O	Mengirim data	-	-
3	RD	I	Menerima data	-	-
4	Terminal R1	-	-	Terminal Resistor	
5	Terminal R2	-	-	Terminal Resistor	
6	RDB+	I/O	-	Mengirim/ menerima data +	Menerima data+
7	SDA-	I/O	-	-	Mengirim data(-)
8	RDA-	-	-	Mengirim/ Menerima data -	Menerima data-
9	SG	-	Signal Ground		

3. USB HOST

NB□□-TW01B dilengkapi dengan port USB HOST, yang merupakan *port* USB A-type. Melalui *port* ini, memori USB dapat dihubungkan untuk melakukan pengunggahan, pengunduhan, operasi proyek dan penyimpanan data terkait. Bentuk dari *port* tersebut ditunjukkan pada Gambar 2.11.



Gambar 2.11 *Port* USB pada Omron NB7W-TW-01B.

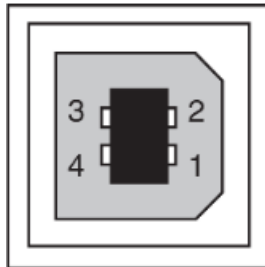
Setiap pin pada port USB tersebut memiliki fungsi masing-masing. Berikut merupakan Tabel 2.4 yang berisikan fungsi kaki pin pada USB pada HMI Omron NB7W-TW01B.

Tabel 2.4 Tabel penjelasan kaki pada USB *Host*.

Pin	Simbol	Fungsi,
1	Vbus	Vbus +5V <i>power supply output</i> (Arus maximal 150mA).
2	D-	Data -
3	D+	Data +
4	GND	Vbus Ground

4. USB SLAVE

Port USB SLAVE adalah *port* USB tipe B, yang dapat menghubungkan *port* USB di sisi PC untuk melakukan pengunggahan, pengunduhan, *debugging* program untuk produk dan pencetakan *Pictbridge*. Untuk pin dari *port* ini ditunjukkan pada gambar 2.12.



Gambar 2.12 *Port* USB SLAVE pada Omron NB7W-TW01B.

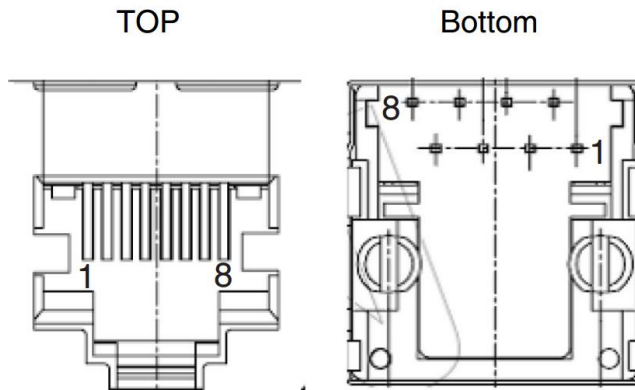
Masing-masing pin dari *port* USB SLAVE juga memiliki fungsi tersendiri. Pada tabel 2.5 menunjukkan fungsi dari pin-pin yang terdapat pada USB SLAVE pada HMI Omron.

Table 2.5 Fungsi dari kaki-kaki pada pin USB SLAVE.

Pin	Simbol	Fungsi
1	Vbus	Vbus
2	D-	Data-
3	D+	Data+
4	GND	Vbus Ground

5. Ethernet *Interface*

NB□□-TW01B dilengkapi dengan antarmuka Ethernet, yang merupakan antarmuka tipe RJ-45 adaptif dengan tingkat transmisi 10M / 100M. *Port* ini dapat dihubungkan dengan PC untuk melakukan pengunggahan dan pengunduhan program, dan juga dapat digunakan untuk berkomunikasi dengan *controller* yang mendukung komunikasi Ethernet. Untuk pin dari *port* ethernet dapat ditunjukkan pada Gambar 2.13.



Gambar 2.13 *Port* Ethernet

Kaki-kaki dari *port* ethernet memiliki fungsi dan kegunaan masing-masing, berikut merupakan fungsi dari kaki pin *port* pada ethernet yang ditunjukkan pada Tabel 2.6.

Tabel 2.6 Fungsi dari kaki-kaki pin ethernet.

Pin	Simbol	Fungsi
1	TD+	Mengirim +
2	TD-	Mengirim Data -
3	RD+	Menerima Data +
4	-	-
5	-	-
6	-	-
7	RD-	Menerima Data -
8	-	-

2.10 NB Designer

Software ini merupakan perangkat lunak yang dapat digunakan untuk mendesain layar *interface* yang di *download* ke HMI (*Human Machine Interface*) sebagai visualisasi untuk berinteraksi dengan mesin. NB Designer merupakan perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data. *Software* ini merupakan keluaran dari Omron yang hanya dapat digunakan untuk HMI Omron seri NB, akan tetapi *software* tersebut dapat digunakan untuk membuat layar HMI bermerek Omron agar dapat mengatur program dari PLC dari vendor lain seperti Siemens, Mitsubishi, dan lain-lain[11].

2.11 Hub

Perangkat ini memiliki peran dalam sebuah jaringan komputer. Hub memiliki kegunaan sebagai perangkat keras penerima sinyal dari suatu komputer dan berfungsi sebagai titik pusat yang menghubungkan semua komputer dalam jaringan tersebut. Hub juga berperan sebagai penguat sinyal kabel UTP, konsentrator dan penyambung. Pada umumnya Hub untuk menghubungkan segmen dari LAN, sebuah hub berisi beberapa *port*. Ketika paket tiba di satu *port*, maka akan disalin ke port lainnya sehingga semua segmen LAN dapat melihat semua paket.

Hub sendiri juga memiliki kelebihan dan kekurangan. Kelebihan menggunakannya yaitu memungkinkan pengguna untuk dapat membagi pada jalur yang sama. Kekurangan menggunakan hub adalah dapat *sharing* jalur yang sama sehingga kecepatan komunikasinya juga harus dibagi dengan hub yang lainnya[5]. Bentuk dari Hub ditunjukkan pada Gambar 2.14.



Gambar 2.14 Hub.

2.12 Relay

Salah satu komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik yaitu relay. Relay merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. Relay biasanya digunakan untuk menggerakkan arus / tegangan yang besar (misalnya peralatan listrik 4 A / AC 220 V) dengan memakai arus / tegangan yang kecil (misalnya 0,1 A / 12 Volt DC). Dalam pemakaiannya biasanya relay yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-paralel dengan lilitannya dan dipasang terbalik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat relay berganti posisi dari on ke off agar tidak merusak komponen di sekitarnya. Konfigurasi dari kontak-kontak relay ada tiga jenis, yaitu *Normally Open* (NO), apabila kontak-kontak tertutup saat relay diberi catu daya, *Normally Closed* (NC) jika kontak terbuka ketika relay terhubung dengan sumber. Berikut merupakan Gambar dari relay. Wujud dari relay ditunjukkan pada Gambar 2.15.



Gambar 2.15 Relay

Penggunaan relay perlu memperhatikan tegangan kontrolernya serta kekuatan relay ketika *switching* arus / tegangan. Biasanya ukurannya tertera pada body relay. Misalnya relay 12VDC/4 A 220V, artinya tegangan yang diperlukan sebagai pengontrolnya adalah 12Volt DC dan mampu *switching* arus listrik (maksimal) sebesar 4 ampere pada

tegangan 220 Volt. Sebaiknya relay difungsikan 80% dari kemampuan maksimalnya agar aman, semakin rendah semakin aman. Relay jenis lain ada yang namanya *reedswitch* atau relay lidi. Relay jenis ini berupa batang kontak terbuat dari besi pada tabung kaca kecil yang dililiti kawat. Pada saat lilitan kawat dialiri arus, kontak besi tersebut akan menjadi magnet dan saling menempel sehingga menjadi saklar yang aktif. Ketika arus pada lilitan dihentikan medan magnet hilang dan kontak kembali terbuka (off).

Penemu relay pertama kali adalah **Joseph Henry** pada tahun 1835. Benda satu ini paling banyak di pake dimobil-mobil sekarang. Kegunaan utamanya adalah meningkatkan efisiensi elektris pada sebuah rangkaian kabel[12].

2.13 Motor DC

Komponen ini adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC. Motor DC atau motor arus searah menggunakan arus langsung dan tidak langsung. *Motor DC* digunakan pada penggunaan khusus dimana diperlukan penyalan torsi yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Berikut merupakan gambar dari motor DC yang ditunjukkan pada Gambar 2.16.



Gambar 2.16 Motor DC

Keuntungan utama motor DC adalah dalam hal pengendalian kecepatan motor DC tersebut, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur tegangan kumparan motor DC, jika tegangan kumparan motor DC ditingkatkan maka kecepatan motor juga akan meningkatkan. Apabila arus medan

diturunkan kecepatan dari motor juga dapat meningkat. Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan rolling mills, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar[13].

2.14 Motor Servo

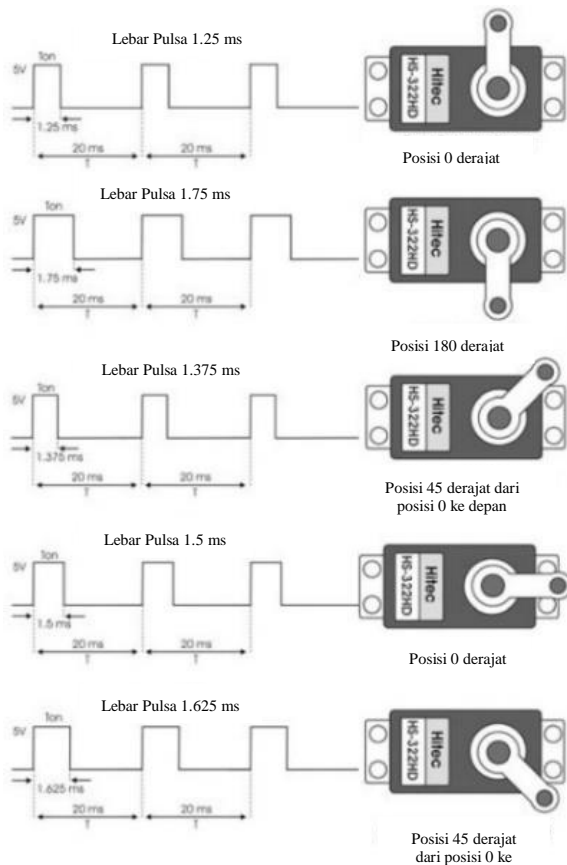
Motor ini merupakan jenis motor DC dengan sistem umpan balik tertutup dengan, serangkaian gear, rangkaian kontrol, dan juga potensiometer. Motor servo sebenarnya didukung oleh komponen-komponen lain yang berada dalam satu paket. Sedangkan fungsi potensiometer dalam motor servo adalah untuk menentukan batas sudut dari putaran servo. Sementara sudut sumbu pada motor servo dapat diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel servo itu sendiri. Oleh karena itu motor servo dapat berputar searah dan berlawanan arah jarum jam. Motor servo dapat menampilkan gerakan 0 derajat, 90 derajat, 180 derajat, hingga 360 derajat. Tak heran jika motor ini banyak diaplikasikan untuk penggerak kaki dan juga lengan robot. Selain itu motor servo juga memiliki torsi yang besar sehingga mampu menopang beban cukup berat. Berikut merupakan bentuk dari motor servo yang ditunjukkan pada Gambar 2.17.



Gambar 2.17 Motor Servo

Prinsip kerja dari motor servo tak jauh berbeda dibanding dengan motor DC yang lain. Hanya saja motor ini dapat bekerja searah maupun berlawanan jarum jam. Derajat putaran dari motor servo juga dapat dikontrol dengan mengatur pulsa yang masuk ke dalam motor tersebut.

Motor servo akan bekerja dengan baik bila pin kontrolnya diberikan sinyal PWM dengan frekwensi 50 Hz. Frekwensi tersebut dapat diperoleh ketika kondisi *duty cycle* berada di angka 1,5 ms. Dalam posisi tersebut rotor dari motor berhenti tepat di tengah-tengah alias sudut nol derajat atau netral. Pada saat kondisi *duty cycle* kurang dari angka 1,5 ms, maka rotor akan berputar berlawanan arah jarum jam. Sebaliknya pada saat kondisi *duty cycle* lebih dari angka 1,5 ms, maka rotor akan berputar searah jarum jam[14]. Berikut skema pulsa kendali motor servo yang ditunjukkan pada Gambar 2.18.



Gambar 2.18 Skema pulsa kendali pada motor servo

2.15 *Limit Switch*

Komponen elektrik sejenis dengan saklar dilengkapi katup yang berfungsi menggantikan tombol. Prinsip kerja *limit switch* sama seperti saklar *push ON* yaitu hanya akan menghubungkan pada saat katupnya ditekan dan akan memutuskan saat katupnya tidak ditekan. *Limit switch* ini termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Berikut bentuk dari *limit switch* yang ditunjukkan pada Gambar 2.19.



Gambar 2.19 *Limit Switch.*

Pada umumnya, *limit switch* digunakan untuk memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain, menghidupkan daya yang besar, serta sebagai sensor posisi atau kondisi dari suatu objek. *Limit switch* memiliki 2 kontak yaitu NO (*Normally Open*) dan kontak NC (*Normally Close*) dimana salah satu kontak akan aktif jika tombolnya tertekan[15].

2.16 *Sensor Inframerah*

Infrared (IR) detektor atau sensor infra merah adalah komponen elektronika yang dapat mengidentifikasi cahaya inframerah. Sensor dibuat khusus dalam satu modul dan dinamakan sebagai IR Detector *Photomodules*. Modul ini terdiri dari sebuah chip detektor inframerah digital yang di dalamnya terdapat *photodiode* dan penguat (*amplifier*). Bentuk dan Konfigurasi Pin *IR Detector Photomodules*. Cara kerja dari sistem sensor infra merah pada dasarnya menggunakan infra merah sebagai media untuk komunikasi data antara *receiver* dan *transmitter*. Sistem akan bekerja jika sinar infra merah yang dipancarkan terhalang

oleh suatu benda yang mengakibatkan sinar infra merah tersebut tidak dapat terdeteksi oleh penerima. Pemancar pada sistem ini terdiri atas sebuah LED infra merah yang dilengkapi dengan rangkaian yang mampu membangkitkan data untuk dikirimkan melalui sinar infra merah, sedangkan pada bagian penerima biasanya terdapat foto transistor, fotodioda, atau inframerah modul yang berfungsi untuk menerima sinar inframerah yang dikirimkan oleh pemancar. Sensor Inframerah ditunjukkan pada Gambar 2.20.



Gambar 2.20 Sensor Inframerah.

Prinsip kerja dari sensor ini Pada rangkaian pemancar hanya pengaturan supaya led infra merah menyala dan tidak kekurangan atau kelebihan daya, oleh karena itu digunakan resistor 680 ohm. Pada rangkaian penerima foto transistor berfungsi sebagai alat sensor yang berguna merasakan adanya perubahan intensitas cahaya infra merah[16]. Pada saat cahaya infra merah belum mengenai foto transistor, maka foto transistor bersifat sebagai saklar terbuka sehingga transistor berada pada posisi *cut off* (terbuka).

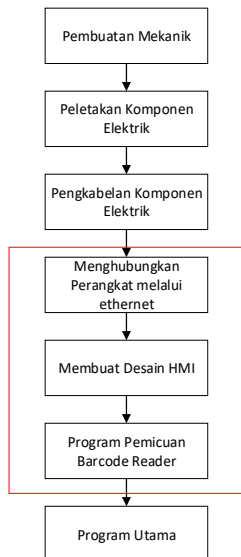
Halaman ini sengaja dikosongkan.

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perencanaan sistem secara keseluruhan dari *Automation Sorting Line System*. Dalam bab ini terdapat tiga sub penjelasan mengenai perancangan sistem, yaitu blok fungsional sistem, perangkat keras dan perangkat lunak. Penjelasan diawali dengan blok fungsional sistem secara keseluruhan dari pembuatan mekanik, pemasangan komponen elektrik dengan pengkabelannya, serta program keseluruhan yang digunakan untuk mengaktifkan sistem tersebut.

3.1 Ruang Lingkup Kerja

Tahap pertama merupakan Pembagian ruang lingkup kerja berupa pembagian pekerjaan yang dilakukan oleh setiap orang dari tahapan awal sampai tahap akhir. Untuk tahapan pembuatan ditunjukkan pada Gambar 3.1.



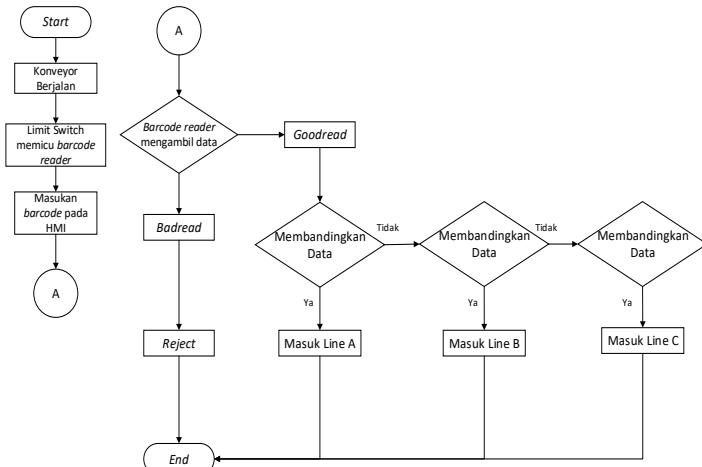
Gambar 3.1 Ruang lingkup kerja.

Pembagian lingkup kerja pada Gambar 3.1 menunjukan proses pembuatan *Automation Sorting Line System* dari awal sampai akhir yang

dimulai dari pembuatan mekanik konveyor pemilah beserta panel, dan dilanjutkan dengan pemasangan komponen elektrik seperti relay, *power supply*, terminal blok, PLC, beserta pengkabelannya. Setelah itu menghubungkan perangkat seperti PLC, HMI, *barcode reader*, dengan menyamakan alamat IP dari perangkat-perangkat tersebut yang dihubungkan melalui *hub switch*, yang kemudian membuat desain tampilan untuk HMI dan juga pemrograman PLC.

3.2 Diagram Alir Keseluruhan

Pada sub bab ini merupakan diagram keseluruhan dari *Automation Sorting Line System* dari awal mesin dinyalakan sampai box yang berhasil disortir ke *sorting line* atau ke konveyor *reject* yang ditunjukkan pada Gambar 3.2.



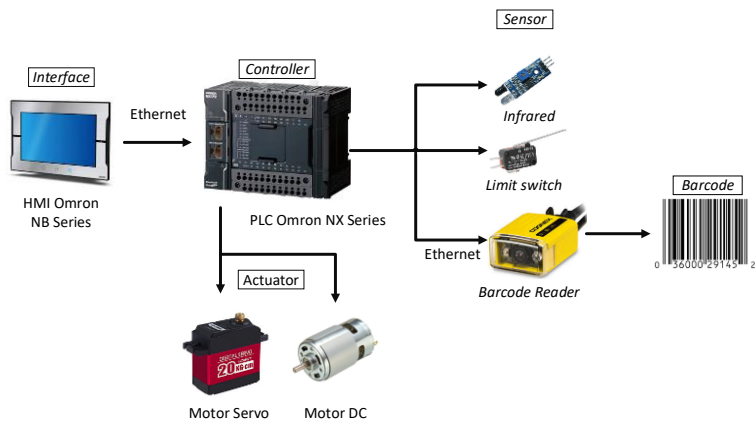
Gambar 3.2 Flowchart sistem keseluruhan

Pada saat mesin dinyalakan seluruh komponen aktif. Jika tombol start pada HMI ditekan maka motor konveyor sudah berputar dan sensor beserta aktuator sudah menyala. Setelah itu pilih barcode yang tersedia pada layar HMI dan masukan pada line yang diinginkan. Ketika box menekan *limit switch* motor pada konveyor akan mati sesaat sampai *barcode reader* berhasil mengambil data berupa digit *barcode* dan dikirim ke PLC. Jika hasilnya *badread*, box akan jalan lurus ke arah konveyor *reject* dan tidak berhasil disortir oleh servo. Jika hasilnya *goodread* dan data berhasil masuk ke PLC box tersebut akan jalan

menuju bagian pemilahan yang kemudian jika datanya sama sesuai dengan masukan pada HMI akan menggerakkan servo untuk mendorong box ke *sorting line* yang ditentukan.

3.3 Perancangan Prangkat Keras

Tahap pada sub bab ini merupakan perancangan perangkat keras yang dilakukan dengan menghubungkan setiap perangkat dengan proses pengkabelan serta ada juga yang menggunakan kabel LAN dengan alamat IP yang disamakan. Hal ini dilakukan agar perangkat-perangkat tersebut dapat bekerja sama untuk menggerakkan sistem ini.



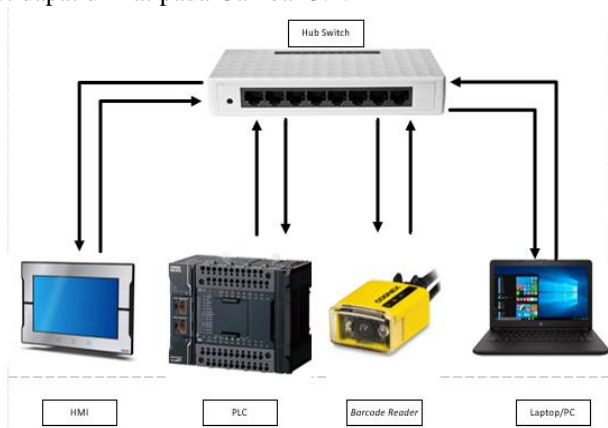
Gambar 3.3 Konfigurasi keseluruhan perangkat keras.

Dilihat dari gambar 3.3 bahwa sensor yang digunakan pada penelitian ini adalah inframerah, *limit switch*, serta *barcode reader* yang digunakan untuk mengambil data dari *barcode* yang melewati *Automation Sorting Line System*. Perangkat-perangkat tersebut merupakan perangkat yang dapat menyusun proses pemilahan pada sistem ini. Data dari *barcode reader* dikirim ke PLC kemudian data tersebut dibandingkan dengan data masukan dari HMI, jika datanya sama maka PLC akan mengirim sinyal ke motor servo agar bergerak untuk menyortir box ke *sorting line* yang disesuaikan dengan masukan di HMI. Untuk pengiriman data dari HMI dan *barcode reader* ke PLC menggunakan komunikasi ethernet yang dihubungkan melalui *hub*. Ketika box sudah masuk ke *sorting line* dan menyentuh *limit switch*, indikator lampu akan menyala dan jumlah dari masuknya box ke *line*

tersebut akan terhitung dan hitungan tersebut ditampilkan di layar HMI pada menu *counter box*.

3.4 Perancangan Sistem Komunikasi

Komunikasi yang digunakan pada sistem *monitoring* ini menggunakan jaringan ethernet dengan wilayah setempat (LAN). LAN terdiri dari medium transmisi bersama dan satu set perangkat keras dan perangkat lunak untuk menghubungkan perangkat ke media dan mengatur akses tertib ke medium. Jaringan LAN merupakan suatu media transmisi bersama serta rangkaian *hardware* dan *software* untuk membuat perangkat saling terhubung. Pada jaringan komunikasi, topologi merupakan istilah yang mengacu pada jalan masuk antar stasiun dari setiap perangkat agar saling terhubung. Topologi jaringan yang digunakan pada *Automation Sorting Line System* berupa topologi *star*. Untuk konfigurasi dari perangkat yang terhubung oleh jaringan ethernet dapat dilihat pada Gambar 3.4.



Gambar 3.4 Konfigurasi perangkat pada jaringan ethernet.

Port LAN dari masing-masing perangkat langsung dihubungkan ke *hub switch* yang berfungsi sebagai node pusat, sehingga *traffic* data mengalir dari tiap node ke node pusat dan kembali lagi[12]. Selain itu jika salah satu kabel dari node terputus, hal tersebut tidak akan mengganggu proses pengiriman data perangkat lainnya, akan tetapi proses pengiriman data dari beberapa perangkat akan terganggu jika hub yang digunakan bermasalah. Cukup mudah untuk mengubah dan menambah komputer ke dalam jaringan yang menggunakan

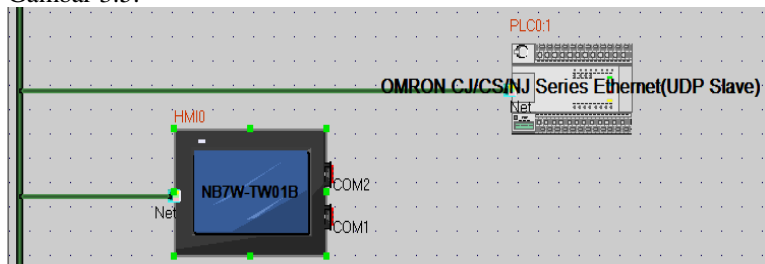
topologi *star* tanpa mengganggu aktivitas jaringan yang sedang berlangsung. Dengan menambah kabel baru dari komputer kita ke lokasi pusat (*central location*) dan pasang kabel tersebut ke *hub*.

3.5 Perancangan Perangkat lunak

Pada bagian ini dibahas perancangan perangkat lunak yang terdiri dari beberapa bagian yaitu pembuatan program PLC untuk memicu sensor agar mengambil data dan mengirimkan ke kontroler, membuat desain tampilan *interface* pada HMI agar operator dapat mengoperasikan mesin, dan *setting* parameter *barcode reader* dengan *Dataman Software* untuk mengetes jarak agar sensor dapat mengambil gambar sera menampilkan hasil data bahwa *barcode reader* berhasil membaca atau tidaknya *barcode* yang lewat agar dapat dikirim ke PC dan PLC.

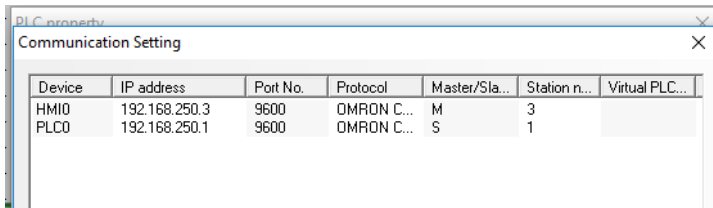
3.5.1 Setup konfigurasi HMI dengan PLC dengan NB Designer

Untuk menghubungkan PLC dengan HMI pada *software* NB Designer, langkah pertama yang harus dilakukan pilih *Programmable Terminal* (PT) Omron NB7W-TW01B dengan PLC CJ/CS/NJ Series. Setelah itu pilih *connector* ethernet dan sambungkan seperti pada Gambar 3.5.



Gambar 3.5 Menghubungkan HMI dengan PLC melalui NB Designer.

Setelah tersambung klik 2 kali pada HMI dan atur alamat IP pada HMI yang akan digunakan. Hal tersebut dilakukan juga pada PLC dengan klik 2 kali pada PLC lalu atur alamat IPnya dan samakan *baudrate* dengan HMI. Untuk mengecek apakah alamat IP antara HMI dengan PLC sudah sama adalah dengan klik pada kabel LAN yang terdapat pada gambar, lalu akan muncul *toolbar* seperti pada gambar 3.6.

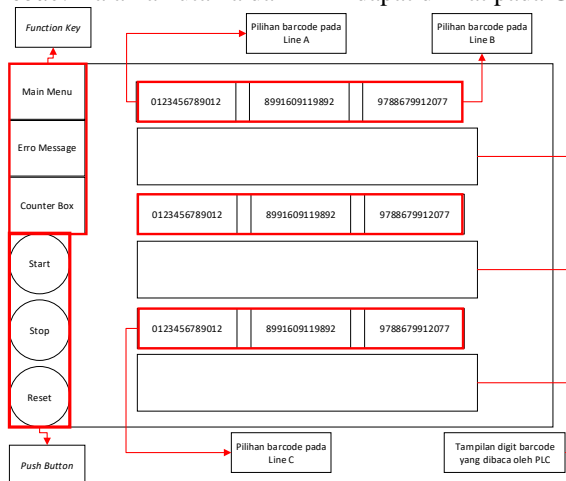


Gambar 3.6 Menyamakan alamat IP HMI dan PLC melalui NB Designer.

Setelah alamat IP dan Baudrate sama, klik *Project Work Space* untuk memulai mendesain tampilan pada HMI yang diinginkan dan digunakan untuk operator menggerakkan suatu mesin.

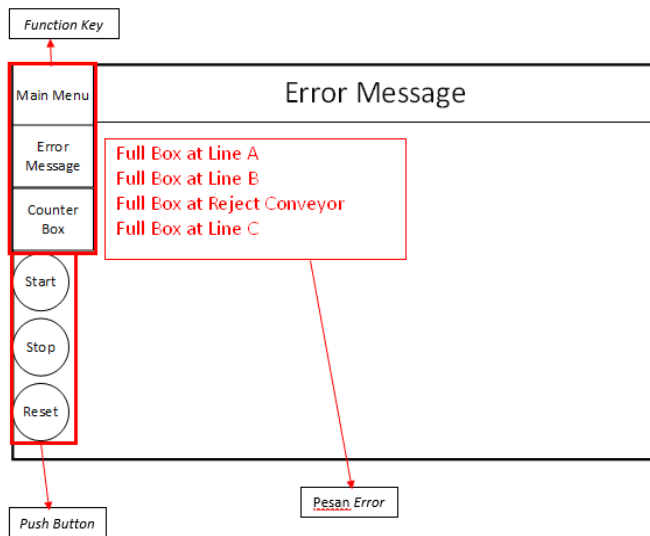
3.5.2 Perancangan HMI

Pada perancangan HMI disini dibuat 3 halaman yang setiap halamannya memiliki fungsi masing-masing. Perancangan ini dibuat melalui *software* NB Designer milik Omron yang hanya dapat digunakan untuk HMI Omron tipe NB saja. Pada halaman utama dari desain yang dibuat berfungsi untuk menekan tombol yang berguna untuk mengatur jalannya sistem dan juga terdapat tombol untuk memasukan digit *barcode*. Halaman utama dari HMI dapat dilihat pada Gambar 3.7.



Gambar 3.7 Halaman Utama pada Sistem *Monitoring*.

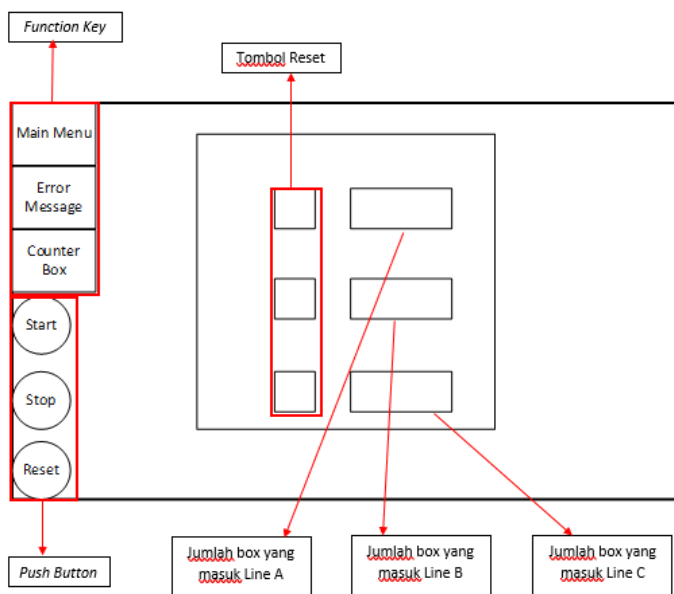
Selain halaman utama terdapat juga halaman yang digunakan untuk menampilkan pesan *error* pada *Automation Sorting Line System*. Halaman *error message* tersebut menampilkan pesan yang dikirim oleh PLC dan menandakan bahwa terjadi ada kesalahan dalam sistem tersebut dan bagian pada sistem *error* tersebut akan ditampilkan pada kolom *message*. Berikut merupakan tampilan dari Halaman *error* pada HMI yang terdapat pada gambar 3.8.



Gambar 3.8 Halaman *Error Message* pada HMI.

Pada halaman *error message* ini dibuat pesan *error* yang muncul ketika terjadi masalah dari bagian mesin. Dengan klik pada menu parts yang terdapat di NB Designer kemudian untuk membuat ukuran layar yang berfungsi untuk menampilkan alarm *error* pada sistem. Selanjutnya pilih menu *alarm setting* yang terdapat pada *project database*. Setelah itu akan muncul tampilan yang akan digunakan untuk menambahkan, mengurangi, serta memodifikasi *address* serta konten *error* yang ingin ditampilkan pada layar HMI apabila terjadi kesalahan pada bagian tertentu yang terdapat pada *Automation Sorting Line System*. Untuk alamat yang digunakan untuk memunculkan PLC menggunakan variabel W yang menunjukkan bit, tetapi variabel W ini tidak akan kesimpan ketika PLC dalam kondisi mati. Jika PLC

dinyalakan lagi maka pesan eror yang muncul pada layar HMI akan terhapus karena variabel tersebut tidak bisa tersimpan dalam PLC ketika kondisi mati. Ketika pada sistem tersebut terjadi kesalahan maka pesan eror tersebut akan muncul di layar HMI pada halaman *error message* sesuai dengan alamat pada PLC. Selain itu terdapat halaman *counter box* yang menampilkan jumlah box yang masuk ke tiap *sorting line* yang ditunjukkan pada Gambar 3.9.



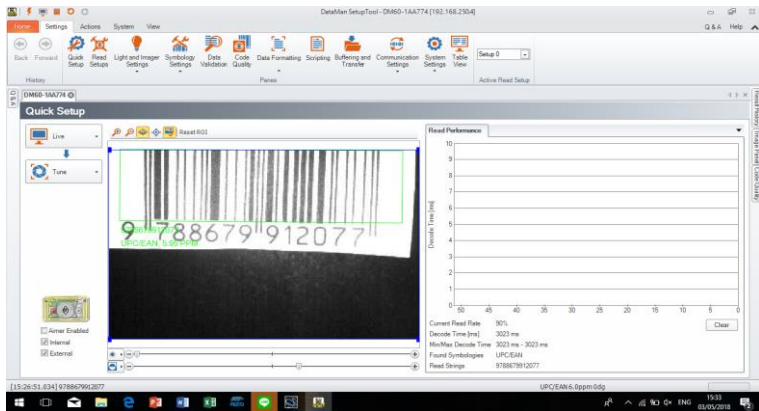
Gambar 3.9 Halaman *Counter Box* pada HMI.

Pada halaman *counter box* terdapat kolom yang dapat menampilkan jumlah box yang berhasil masuk ke *sorting line*. Pada halaman ini dibuat *number display* yang berfungsi untuk menampilkan jumlah dari box yang masuk ke *sorting line*. Box tersebut akan dihitung ketika telah turun dan mengaktifkan *limit switch* pada tiap *sorting line*. Untuk alamat pada masing-masing *number display* diberi dengan variabel D sebagai *data memori*, karena variabel D dapat digunakan dalam satuan *word*, dan nilai yang telah tersimpan tidak akan hilang walaupun PLC dalam kondisi mati. Pada halaman tersebut juga diberi tombol *reset* yang digunakan untuk mengulang jumlah dari penghitung

box kembali ke angka 0. Masing-masing dari tombol reset tersebut digunakan untuk mereset jumlah box yang masuk ke tiap *sorting line*.

3.5.3 *Setting Parameter Reader ke PC melalui Dataman Software*

Agar hasil pembacaan dari sensor *barcode* dapat dilihat, perlu dihubungkannya *barcode reader* tersebut ke PC melalui jaringan ethernet. Jika *barcode reader* sudah aktif, buka *software* dataman untuk menghubungkan kedua perangkat tersebut. IP *address* dari kedua perangkat harus disamakan terlebih dahulu. Jika sudah disamakan, kedua perangkat tersebut sudah terhubung dan menampilkan berbagai menu yang digunakan untuk mengatur kendali dari sensor DM60.



Gambar 3.10 Tampilan menu *Quick Setup* pada *Dataman Software*.

Pada Gambar 3.10 merupakan tampilan dari menu *Quick Setup* yang menampilkan gambar dari pembacaan *barcode*. Menu tersebut merupakan tampilan paling awal ketika laptop atau PC berhasil terhubung dengan *barcode reader*. Pada menu tersebut terdapat tombol *Live*, *Tune*, dan *Test* yang masing-masing memiliki fungsi tersendiri. Tombol *Live* memiliki fungsi untuk membuat lensa menampilkan gambar yang diambil secara langsung. Lalu tombol *Tune* memiliki fungsi untuk mengambil gambar dari *barcode* dan memunculkan hasil berupa *read* yang menunjukkan bahwa sensor berhasil membaca atau *no read* yang berarti bahwa sensor gagal membaca.

3.5.4 Membuat Modul *Input Output Assembly*

Dengan membuat tipe data untuk modul *input* dan *output assembly* yang digunakan untuk memicu *barcode reader* agar dapat memulai proses akuisisi data. Bagian *input assembly* digunakan sebagai masukan oleh PLC atau hanya digunakan sebagai indikator. Berikut merupakan nama-nama dari tipe data beserta fungsinya yang digunakan untuk memicu *barcode reader*:

Tabel 3.1 Modul *input assembly*.

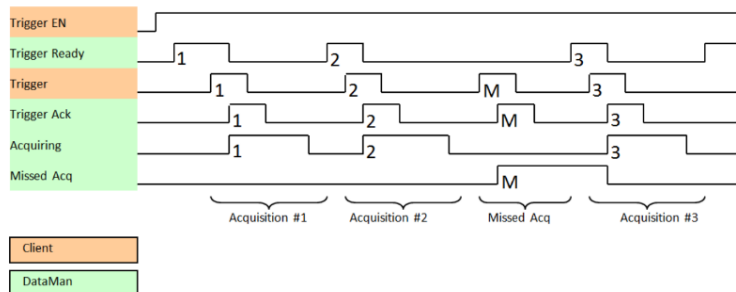
Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11	0	Reserved				Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	General Fault	Reserved			Results Available	Results Buffer Overrun	Decode Complete Toggle	Decoding
	2	Soft Event Ack 7	Soft Event Ack 6	Soft Event Ack 5	Soft Event Ack 4	Soft Event Ack 3	Soft Event Ack 2	Soft Event Ack 1	Soft Event Ack 0
	3-5	Reserved							
	6	Trigger ID (16-bit integer)							
	7								
	8	Result ID (16-bit integer)							
	9								
	10	Result Code (16-bit integer)							
	11								
	12	Result Extended (16-bit integer)							
	13								
	14	Result Data Length (16-bit integer)							
	15								
	16	Result Data 0							
	...								
	499	Result Data 483							

Tabel 3.2 Modul *output assembly*.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Reserved				Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	2	Reserved							
	3								
	4								
	5	User Data Option (16-bit integer)							
	6	User Data Length (16-bit integer)							
	7								
	8								
		User Data 0							
	...								
	499	User Data 491							

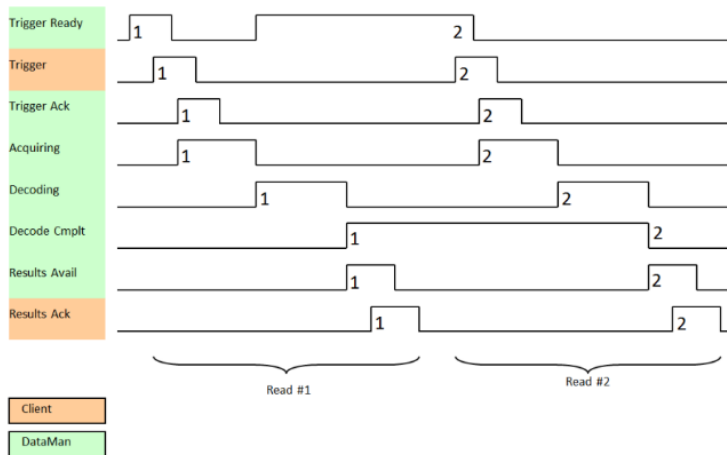
Sebelum mengirim bit-bit tersebut dari sysmac studio, sebaiknya *copy* EDS (*Electronic Data Sheet*) dari *barcode reader* yang digunakan ke file EIP (Ethernet IP) Connection pada file Sysmac studio. Setelah itu, isi target *device* pada menu *connection* yang terdapat di sysmac studio, pastikan alamat IP yang diisi sesuai dengan alamat IP dari *barcode reader* yang digunakan.

Selanjutnya adalah membuat *function block* program PLC menggunakan *structure text*. *Function Block* tersebut memiliki fungsi untuk memicu *barcode reader* agar dapat mengambil gambar dari *barcode* dan kemudian hasil tersebut di proses sesuai dengan *Timing Chart* yang ada dibawah ini. Pada proses ini, bit dari Trigger Enable harus diaktifkan. Jika bit *trigger ready* dalam keadaan aktif, kemudian bit *Trigger* diaktifkan, maka sensor akan memulai menembakan *flash* untuk mengambil data. Jika kondisi *trigger ready* masih tidak aktif, tetapi PLC sudah mengaktifkan bit *trigger*, maka hasilnya akan *missed Acq* atau *barcode reader* tidak menembakan *flash*. Berikut merupakan *Timing Chart* dari proses akuisisi yang ditunjukkan pada Gambar 3.11.



Gambar 3.11 *Timing Chart Acquisition Sequence*

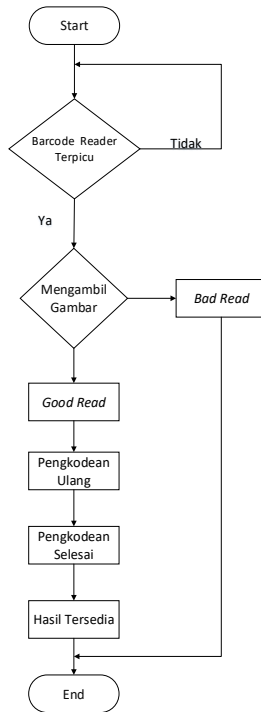
Sesuai dengan *Timing chart* di atas, yang dilakukan adalah memberi pulse dari PLC (*client*), sementara untuk proses yang dilakukan dataman berjalan secara otomatis dan hanya berfungsi sebagai indikator dari jalannya proses akuisisi data. Selain proses akuisisi juga terdapat *timing chart* yang berisikan proses jika dataman berhasil memperoleh data (Result Sequence) dari *barcode* yang telah didapatkan. Proses tersebut dapat dilakukan jika barcode reader berhasil terpicu oleh sinyal dari bit *Trigger Enable* yang dikirim oleh PLC. Proses tersebut terdapat pada Gambar 3.12.



Gambar 3.12 Proses *Result Sequence*.

Timing Chart diatas menunjukkan proses yang terjadi ketika dataman berhasil memperoleh (*Acquiring*) data. Jika bit acquiring aktif ketika dataman memperoleh data, langkah selanjutnya data tersebut akan di pengkodean ulang (*decoding*) oleh dataman itu sendiri, dan jika proses pengkodean selesai, maka bit decoding akan mati dan bit decode complete akan aktif yang menunjukkan bahwa proses pengkodean telah selesai dilakukan.

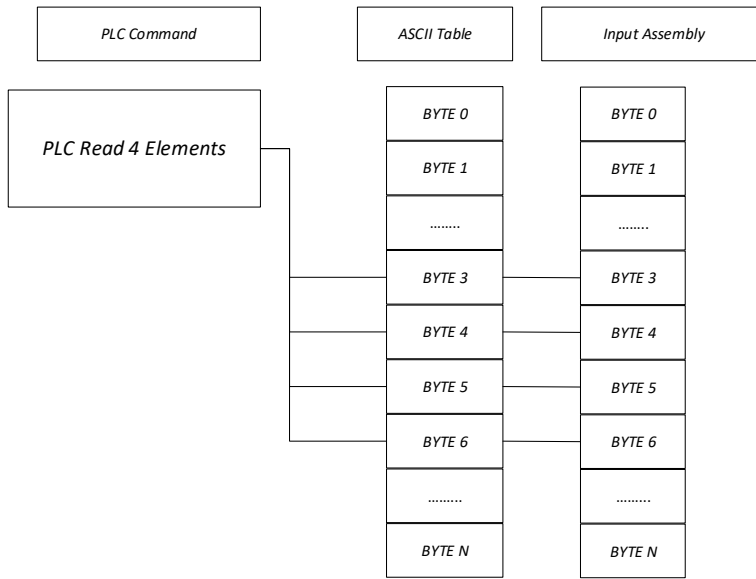
Selain itu terdapat keluaran yang digunakan sebagai indikator pada saat pengambilan data. Indikator tersebut berupa *goodread* dan *badread*. Ketika *barcode* berhasil terbaca dan mendapatkan data, indikator lampu hijau pada *barcode reader* akan berkedip dan akan muncul angka dari digit *barcode* pada menu *result history* pada Dataman *software*, serta data akan muncul pada program yang telah dibuat di Sysmac studio. Jika *barcode reader* tidak berhasil membaca, maka data tidak muncul pada program PLC dan lampu merah pada *barcode reader* akan berkedip. Berikut merupakan *flow chart* Dataman dari proses awal ketika Dataman dipicu oleh PLC sampai data diterima PLC dan juga *barcode reader* yang tidak merespon ketika diberi pemicu oleh PLC.



Gambar 3.13 Flowchart proses pengambilan data oleh *barcode reader*

Pada Gambar 3.13 menunjukkan proses pengambilan data pada pembacaan *barcode* yang dilakukan oleh *barcode reader*. Sensor tersebut akan menembakan flash untuk mengambil gambar jika berhasil dipicu oleh PLC. Jika hasilnya *goodread* data *barcode* tersebut akan dikirim ke kontroler dan indicator berhasilnya sensor tersebut dalam mengambil data adalah berkedipnya led hijau pada *reader* tersebut. Jika hasilnya *badread* indicator yang muncul dari sensor tersebut merupakan lampu merah yang berkedip.

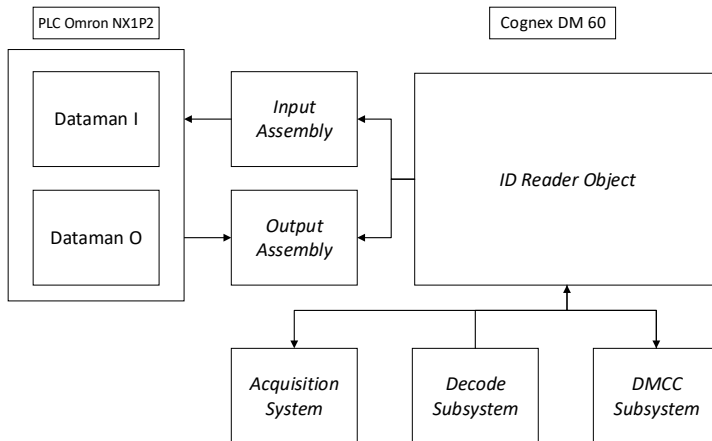
Perintah dari Sysmac disusun untuk bekerja pada tabel data yang ada di PLC. Setiap tabel data adalah *array* dari tipe data yang diberikan (*Byte, int, float, dll*). Perintah digunakan untuk membaca / menulis satu atau lebih item data dari tabel data tertentu. Item dialamatkan dengan menentukan tabel data dan indeks item dalam tabel (basis indeks dari 0). Pembacaan data dari PLC dari Dataman ditunjukkan pada Gambar 3.14.



Gambar 3.14 Pembacaan data PLC dari DataMan.

Pada Gambar 3.14 proses pengkodean Result Data dimulai pada *byte offset* 3(dihitung dari 0). Untuk membaca 4 *byte* pertama data hasil, pengguna harus mengeluarkan perintah melalui PLC seperti pada gambar 3.11. Pada proses pemicuan antara PLC dengan DataMan *Barcode Reader* terjadinya pengiriman pesan tersirat yang mengirimkan data I / O spesifik waktu-kritis aplikasi, dan bisa *point-to-point* atau *multicast*. Pesan ini membutuhkan respons dari perangkat penerima.

Pemicuan dengan cara mengirimkan sinyal dari setiap bit pada PLC melalui Ethernet IP memungkinkan *input* dan *output* pembaca Dataman diletakkan ke dalam *tag set* pada Sysmac Studio PLC Omron. Untuk alamat IPnya harus disesuaikan dengan perangkat yang dituju. Setelah koneksi ini dibuat, data ditransfer secara siklus pada interval yang ditentukan pengguna yang biasanya berkisar antara 10 ms, 50 ms, 100ms, dll. berikut merupakan Gambar 3.15 yang menunjukkan alur pemicuan *point to point* antara *barcode reader* dengan PLC.

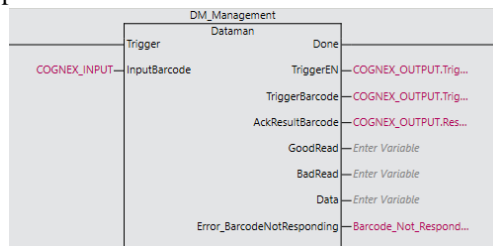


Gambar 3.15 Alur Pemicuan Barcode reader dengan PLC.

Modul *input output assembly* dapat dipanggil untuk memproses akuisisi data *barcode* dengan *function block* yang berisikan program pada PLC untuk memicu *barcode reader* agar dapat melakukan proses akuisisi.

3.5.5 Function Block Dataman

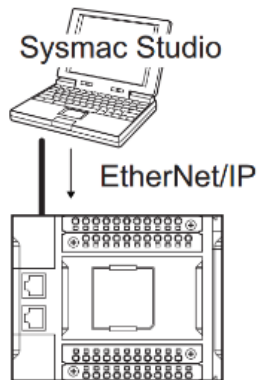
Agar *barcode reader* dapat bekerja, dibuatlah *function block* berisikan program berbasis *structure text*. *Function Block* tersebut akan aktif ketika mendapat trigger dari luar yaitu dari Limit switch. Hasil keluaran dari program ini akan masuk pada *function block* yang digunakan untuk membandingkan data dari *barcode reader* dengan HMI yang berfungsi untuk menggerakkan motor servo untuk mendorong box ke *sorting line* yang dituju. Bentuk dari *function block* tersebut ditunjukkan pada Gambar 3.16.



Gambar 3.16 Fuction Block Dataman

3.5.6 Konfigurasi Sysmac Studio pada PC dengan Omron NX1P2

Untuk menghubungkan langsung antara PC dengan Omron NX1P2 diatur dengan menggunakan Sysmac Studio. Untuk alamat IP dan perangkat tidak perlu ditentukan. Pastikan bahwa lampu indikator warna oranye pada CPU Omron NX1P2 menyala dan berkedip yang menandakan bahwa PLC tersebut terhubung dengan perangkat lain. Selain menghubungkan langsung juga dapat melalui Hub.



Gambar 3.17 Menghubungkan Sysmac Studio dengan ethernet IP.

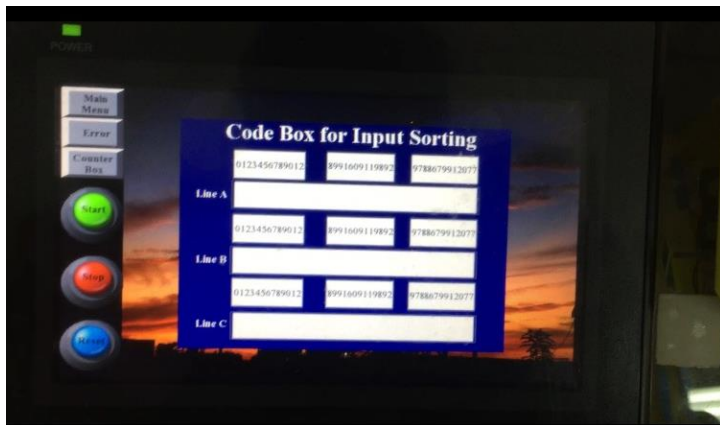
BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan pengambilan data dan pengujian sistem beserta analisisnya. Pengambilan data berupa hasil penerimaan data *barcode* pada PLC yang dikirim oleh DM60 yang akan dibandingkan dengan kode berupa angka yang merupakan masukan dari HMI dan jika data dari sensor sama dengan masukan dari HMI, perbandingan kecepatan pemicuan *barcode reader* dari PLC dengan sensor eksternal dan kecepatan pemicuan manual oleh PC ke barcode reader dengan mode manual yang terdapat pada Dataman *software*.

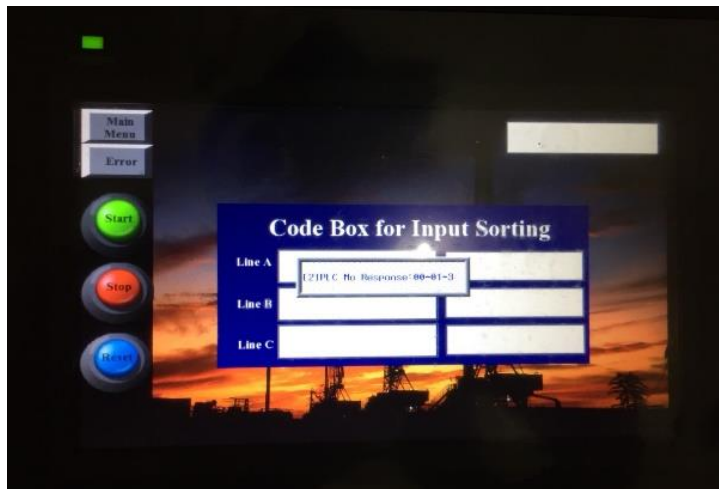
4.1 Pengujian Human Machine Interface Omron NB7W-TW01B

Pada Pengujian HMI ini ditujukan untuk mengetahui PLC Omron NX1P2 dapat berkomunikasi baik dengan HMI., serta dapat melakukan perintah perintah untuk pengendalian dan *Monitoring* pada *Automation Sorting Line System*. Tampilan HMI Omron yang terhubung pada PLC Omron NX1P2 dapat dilihat pada gambar. Tahapan yang pertama kali dalam pengujian HMI Omron NB7W-TW01B yaitu mengunduh program pada PLC Omron, selanjutnya hubungkan HMI Omron NB7W dengan catu daya 24 VDC. Setelah itu samakan alamat IP antara HMI dengan PLC, dan selanjutnya hubungkan dengan kabel LAN.



Gambar 4.1 Human Machine Interface pada Automation Sorting Line System

Hasil pengujian Omron NB7W-TW01B dilakukan dengan cara menghubungkan HMI dengan PLC menggunakan ethernet IP. Jika perangkat PLC dengan HMI terhubung, maka indikator lampu pada PLC menyala berwarna hijau dan berkedip, dan jika perangkat tidak terhubung maka akan muncul “PLC no Respon” pada layar HMI seperti yang ditunjukkan pada gambar berikut ini.



Gambar 4.2 *Human Machine Interface* (HMI) ketika belum terhubung dengan PLC.

4.2 Pengujian Pembacaan *Input Text* HMI dan *push button* pada HMI

Pengujian ini digunakan untuk mengetahui hasil masukan dari HMI yang harus dapat terbaca pada PLC. Dengan memasukkan nilai dari *barcode* pada HMI, nilai tersebut harus terbaca dan masuk pada PLC dengan kode yang sesuai dengan masukan pada HMI. Tahapan yang pertama adalah mengunduh program yang dibuat untuk HMI pada NB Designer. Selanjutnya mengubah mode online pada sysmac studio agar dapat terhubung dengan PLC. Setelah itu klik menu watch pada Sysmac studi dan masukan nama *Code HMI A*, *Code HMI B*, *Code HMI C* yang telah dibuat pada global variable Sysmac studio. Kemudian masukan angka/huruf yang diinginkan pada *Input Text* HMI yang diinginkan,

maka hasil tersebut akan terbaca pada menu *watch* yang terdapat pada Sysmac Studio.

Peralatan yang dibutuhkan untuk pengujian ini adalah *power supply* yang harus terhubung pada HMI, PLC, Sysmac Studio yang digunakan untuk membaca hasil masukan dari HMI serta kabel LAN yang digunakan untuk menghubungkan perangkat-perangkat yang digunakan seperti HMI, PLC, serta Laptop. Hasil dari pengujian komunikasi data integer antara *Human Machine Interface* dengan kontroler pada sistem *monitoring* ini dapat dilihat pada tabel 4.2, tabel 4.3, dan juga pada tabel 4.4.

Tabel 4.1 Hasil pengujian komunikasi pada *line A*.

Line A (HMI)	Code HMI A (PLC)
9778679912077	9778679912077
0123456789012	0123456789012
8911609119892	8911609119892
9781855682979	9781855682979

Pada Tabel 4.1 menampilkan angka dari HMI pada *line A* yang berhasil dibaca oleh PLC. Angka tersebut dilihat pada menu *watch* dari *software* Sysmac Studio milik Omron. Alamat dari kedua perangkat tersebut harus sesuai. Pada Sysmac Studio alamat tersebut harus terdaftar di global variabel.

Tabel 4.2 Hasil pengujian komunikasi pada *line B*.

Line B (HMI)	Code HMI B (PLC)
9778679912077	9778679912077
0123456789012	0123456789012
8911609119892	8911609119892
9781855682979	9781855682979

Pada Tabel 4.2 menampilkan angka dari HMI pada *line B* yang berhasil dibaca oleh PLC. Angka tersebut dilihat pada menu *watch* dari *software* Sysmac Studio milik Omron. Alamat dari kedua perangkat tersebut harus sesuai. Pada Sysmac Studio alamat tersebut harus terdaftar di global variabel.

Tabel 4.3 Hasil pengujian komunikasi pada *line B*.

Line C (HMI)	Code HMI C (PLC)
9778679912077	9778679912077
0123456789012	123456789012
8911609119892	8911609119892
9781855682979	9781855682979

Pada Tabel 4.3 menampilkan angka dari HMI pada *line B* yang berhasil dibaca oleh PLC. Angka tersebut dilihat pada menu *watch* dari *software* Sysmac Studio milik Omron. Alamat dari kedua perangkat tersebut harus sesuai. Pada Sysmac Studio alamat tersebut harus terdaftar di global variabel.

Tabel 4.4 Pengujian penekanan tombol pada HMI.

	Mesin	Lampu Indikator
Start	Menyala	Menyala
Stop	Mati	Berkedip
Reset	Mati	Menyala

Pada Tabel 4.4 dilakukan pengujian pada tombol pada HMI yang berfungsi untuk menggerakkan sistem. Jika tombol start pada HMI ditekan motor pada konveyor mulai menyala dan lampu indikator warna hijau juga menyala. Jika tombol stop pada HMI ditekan motor pada konveyor mati dan lampu indikator menyala warna merah. Saat tombol reset ditekan sistem masih keadaan mati, lampu indikator menyala warna kuning, dan sistem siap untuk dijalankan ketika tombol start ditekan.

4.3 Pengujian pembacaan *Barcode* di *software* Sysmac Studio

Pada bagian ini dilakukan pengujian komunikasi dari hasil yang diambil oleh *barcode reader* dan dikirim pada PLC yang terbaca pada Sysmac Studio. Dengan hasil yang berhasil diambil oleh *barcode reader* setiap digit dari *barcode* tersebut harus terbaca pada modul *input assembly* “Result Data” yang dapat dilihat pada menu *watch* di Sysmac Studio. Selain dari digit *barcode*, Panjang dari digit *barcode* pada *input assembly* “Result Length” yang terbaca pada PLC harus sesuai dengan data yang diambil oleh *barcode reader*.

Peralatan yang dibutuhkan untuk pengujian ini komunikasi *barcode reader* dengan PLC yaitu *barcode reader* yang digunakan untuk mengambil data, PLC Omron sebagai penerima data yang diambil oleh *reader*, *software* Sysmac Studio yang digunakan untuk melihat data yang berhasil diterima oleh PLC, serta kabel LAN yang digunakan untuk menghubungkan laptop, PLC, serta *barcode reader*. Data dari barcode yang berhasil diambil terdapat pada Tabel 4.5, Tabel 4.6, Tabel 4.7, Tabel 4.8.

Tabel 4.5 Hasil baca dari *barcode reader* yang dikirim pada PLC dengan *barcode* 9778679912077.

Result Length	13	
Result Data	Hexadecimal	ASCII
Result Data [0]	39	9
Result Data [1]	37	7
Result Data [2]	37	7
Result Data [3]	38	8
Result Data [4]	36	6
Result Data [5]	37	7
Result Data [6]	39	9
Result Data [7]	39	9
Result Data [8]	31	1
Result Data [9]	32	2
Result Data [10]	30	0
Result Data [11]	37	7
Result Data [12]	37	7

Pada Tabel 4.5 didapat hasil dari *barcode* 9778679912077 yang berhasil dibaca oleh PLC. Setiap *byte Result Data* menampilkan 1 digit *barcode* yang telah didapat. Kemudian pada *Byte Result Length* menampilkan Panjang dari data *barcode* yang berhasil diambil.

Tabel 4.6 Hasil baca dari *barcode reader* yang dikirim pada PLC dengan *barcode* 0123456789012.

Result Length	13	
Result Data	Hexadecimal	ASCII
Result Data [0]	30	0
Result Data [1]	31	1
Result Data [2]	32	2
Result Data [3]	33	3

Result Data	Hexadecimal	ASCII
Result Data [4]	34	4
Result Data [5]	35	5
Result Data [6]	36	6
Result Data [7]	37	7
Result Data [8]	38	8
Result Data [9]	39	9
Result Data [10]	30	0
Result Data [11]	31	1
Result Data [12]	32	2

Pada Tabel 4.6 didapat hasil dari *barcode* 0123456789012 yang berhasil dibaca oleh PLC. Setiap *byte Result Data* menampilkan 1 digit *barcode* yang telah didapat. Kemudian pada *Byte Result Length* menampilkan panjang dari data *barcode* yang berhasil diambil.

Tabel 4.7 Hasil baca dari *barcode reader* yang dikirim pada PLC dengan *barcode* 8911609119892.

Result Length	13	
Result Data	Hexadecimal	ASCII
Result Data [0]	38	8
Result Data [1]	39	9
Result Data [2]	31	1
Result Data [3]	31	1
Result Data [4]	36	6
Result Data [5]	30	0
Result Data [6]	39	9
Result Data [7]	31	1
Result Data [8]	31	1
Result Data [9]	39	9
Result Data [10]	38	8
Result Data [11]	39	9
Result Data [12]	32	2

Pada Tabel 4.7 didapat hasil dari *barcode* 8911609119892 yang berhasil dibaca oleh PLC. Setiap *byte Result Data* menampilkan 1 digit *barcode* yang telah didapat. Kemudian pada *Byte Result Length* menampilkan panjang dari data *barcode* yang berhasil diambil.

Tabel 4.8 Hasil baca dari *barcode reader* yang dikirim pada PLC dengan *barcode* 9781855682979

Result Length	13	
Result Data	Hexadecimal	ASCII
Result Data [0]	39	9
Result Data [1]	37	7
Result Data [2]	38	8
Result Data [3]	31	1
Result Data [4]	38	8
Result Data [5]	35	5
Result Data [6]	35	5
Result Data [7]	36	6
Result Data [8]	38	8
Result Data [9]	32	2
Result Data [10]	39	9
Result Data [11]	37	7
Result Data [12]	39	9

Pada Tabel 4.6 didapat hasil dari *barcode* 0123456789012 yang berhasil dibaca oleh PLC. Setiap *byte Result Data* menampilkan 1 digit *barcode* yang telah didapat. Kemudian pada *Byte Result Length* menampilkan panjang dari data *barcode* yang berhasil diambil. Berikut merupakan gambar dari hasil yang didapat dan ditampilkan pada menu watch yang terdapat di Sysmac Studio.

Result_Length	13
▼ Result_Data[0-478]	
Result_Data[0]	9 (16#39)
Result_Data[1]	7 (16#37)
Result_Data[2]	8 (16#38)
Result_Data[3]	8 (16#38)
Result_Data[4]	6 (16#36)
Result_Data[5]	7 (16#37)
Result_Data[6]	9 (16#39)
Result_Data[7]	9 (16#39)
Result_Data[8]	1 (16#31)
Result_Data[9]	2 (16#32)
Result_Data[10]	0 (16#30)
Result_Data[11]	7 (16#37)
Result_Data[12]	7 (16#37)

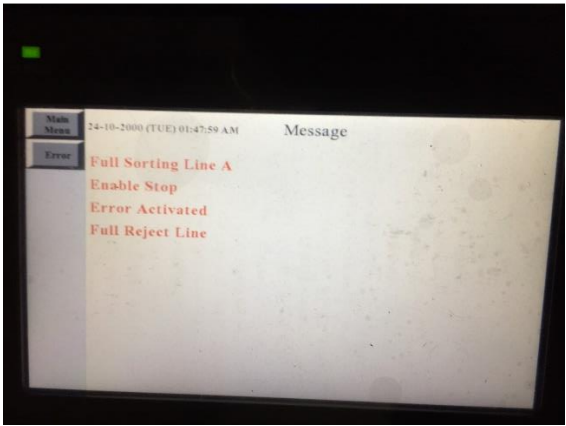
Gambar 4.3 Data *Barcode* 9788679912077 yang berhasil dibaca Oleh PLC.

4.4 Pengujian Memunculkan *Error message* pada HMI

Pengujian Human Machine Interface Omron NB7W-TW01B ini digunakan untuk mengetahui PLC Omron NX1P2 dapat memunculkan pesan eror pada Tampilan HMI Omron yang terhubung pada PLC Omron NX1P2 dapat dilihat pada gambar. Jika terjadi suatu eror pada bagian dari mesin, maka *address* pada PLC tersebut akan ditampilkan pada HMI berupa pesan. Peralatan yang dibutuhkan yaitu HMI Omron yang digunakan untuk menampilkan pesan eror yang terjadi pada system, PLC Omron, kabel LAN yang digunakan untuk menghubungkan PLC dengan HMI. Pengujian data tersebut terdapat pada Tabel 4.9.

Tabel 4.9 Pengujian Status pada HMI

Error Message	Status	Indikator
Full Box Sorting Line A	Muncul	Lampu Nyala
Full Box Sorting Line B	Muncul	Lampu Nyala
Full Box Sorting Line C	Muncul	Lampu Nyala
Full Box Sorting Line A,B	Muncul	Lampu Nyala
Full Box Sorting Line A,C	Muncul	Lampu Nyala
Full Box Sorting Line B,C	Muncul	Lampu Nyala
Full Box Sorting Line A,B,C	Muncul	Lampu Nyala
Motor Stop	Muncul	Motor mati



Gambar 4.4 Terjadinya *full box* pada *sorting line A*.

Pesan pada Gambar 4.4 tersebut muncul dikarenakan penuhnya box yang berada pada tiap *Sorting Line* dan *line reject* yang menyebabkan sensor aktif selama beberapa detik akibat mendeteksi box yang masih belum diambil, sehingga dengan kejadian tersebut mesin akan mati dan tidak bisa berjalan, dan kemudian memunculkan pesan pada HMI.

4.5 Pengujian Proses Pengiriman Data

Pengambilan data pada proses ini dilakukan agar mengetahui *delay* dari proses pengiriman data dari satu node ke node lain. Pada pengujian ini dibutuhkan *barcode reader* sebagai node yang digunakan untuk menerima atau mengirim data ke laptop/pc, *software wireshark* yang digunakan untuk membaca proses pengiriman data yang berlangsung, Dataman software yang digunakan untuk memicu *barcode reader* melalui laptop serta kabel LAN. yang digunakan untuk membaca pengiriman data melalui ethernet yang telah terhubung oleh hub. Berikut rumus yang digunakan untuk menghitung *delay*.

$Delay = \text{Waktu Paket Diterima} - \text{Waktu Paket Dikirimkan}$

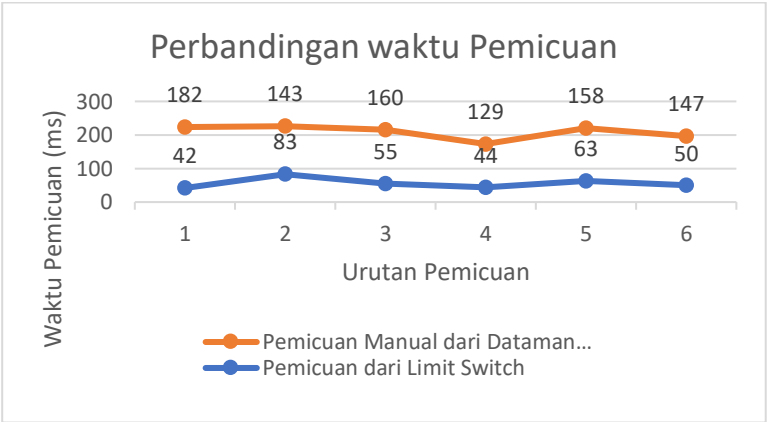
Tabel 4.10 Data yang digunakan untuk mencari *delay*.

	<i>Data Length (Bytes)</i>	<i>Time (s)</i>	<i>Delay (s)</i>
Dikirim	201	9,428359000	0,033911
Diterima	169	9,462270000	
Dikirim	1514	9,754233000	0,000131
Diterima	54	9,754364000	
Dikirim	201	9,104442000	0,0007
Diterima	54	9,105142000	
Dikirim	598	8,158367000	0,0499
Diterima	169	8,208267000	
Dikirim	582	9,350371000	0,064774
Diterima	169	9,410145000	

Pada Tabel 4.10 terdapat panjang data berupa *byte* yang dikirim oleh suatu node dan diterima oleh node lain. Pada proses pengiriman tersebut terdapat *delay* dari setiap pengiriman data oleh suatu node. Waktu *delay* tersebut memiliki satuan *milisecond*. *Delay* tersebut dicari melalui selisih waktu data yang diterima dengan data yang dikirim.

4.6 Pengujian proses waktu pemicuan

Pada pengujian ini dilakukan perbandingan waktu untuk memicu *barcode reader* secara manual dengan memberi pulse melalui *Dataman software*. Hasil dari pengujian tersebut dibandingkan dengan waktu pemicuan ketika box berjalan dan menyentuh *limit switch* yang digunakan untuk memicu *barcode reader*. Untuk mengetahui waktu pemicuan yang telah dilakukan, digunakan software *Dataman* yang menampilkan waktu pemicuan dengan satuan (ms). Untuk pemicuan manual dilakukan dengan menghubungkan langsung laptop/PC dengan barcode reader, kemudian untuk trigger melalui PLC dengan menghubungkan semua perangkat pada hub. Waktu pemicuan tersebut ditampilkan pada Grafik 4.8.



Gambar 4.5 Grafik waktu pemicuan

BAB V

PENUTUP

Bab penutup ini berisi mengenai kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya. Dari pengujian dan analisa Perancangan Komunikasi dari Sistem *Monitoring* pada *Automation Sorting Line System* dapat diperoleh beberapa kesimpulan seperti Cognex DM60 dapat dipicu melalui PLC dengan variabel tanpa menggunakan *memory address* seperti %W,%D, dan juga %H. Selain itu, ketika ingin memicu DM60 *Barcode reader* bit *Trigger Enable* harus didaftarkan pada *tag set*, dan sebelum pemicuan harus diaktifkan agar dapat memulai proses akuisisi. Kemudian untuk HMI NB7W-TW01B dapat digunakan untuk menampilkan pesan eror yang terjadi pada mesin. Pada saat proses pengiriman data dari masing-masing perangkat jika terkena *noise*, maka menimbulkan pengiriman data yang tidak lengkap dan *delay* pada saat pengiriman. Untuk saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini yaitu melakukan *setup* terlebih dahulu dalam menggunakan Cognex DM60 *Barcode Reader*. Lalu dalam membuat desain tampilan pada HMI sebaiknya hindari *background* yang memiliki ukuran gambar yang tinggi karena akan membuat lamanya proses penekanan pada layar. Sebaiknya mengecek kondisi kabel LAN yang digunakan pada setiap perangkat, karena jika keadaannya tidak rapat dapat mengganggu pengiriman data antar perangkat.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] R.Todd Swinderman. "7th Edition Belt Conveyor for Bulk Materials". Prepared by The Engineering Conference with Metric.
- [2] Chikezie Nwahoa, Michael D. Holloway, Oliver A. Onyewuenyi. "Process Plant Equipment : Operation, Control, and Reliability".
- [3] Prof..G.B.Firame, Priyanka P. Banderwar, Shreeya V. Kulkarni, Swati R. Bhosale. "Automatic Box Sorting Machine". International Journal for scientific Research & Development Vol.4, Issue 04, 2016.
- [4] Cognex Dataman Reference Manual. Version 5.7.0.102
- [5] William Stallings, "Data Computer and Communication". Upper Saddle River, New Jersey 07458.
- [6] Bayindir R, Centinceviz Y.A water pumping control system with a programmable logic controller (PLC) and industrial wireless modules for industrial plants-An experimental setup. ISA Trans 2011;50;321-8.
- [7] Saad N, Arrofiq M. A PLC based modified fuzzy controller for PWM-driven induction motor drive with constant V/Hz ratio control. Robot Computer Integrated Manuf 2012;28(2):95-112.
- [8] Omron, NX Series NX1P2 CPU Unit Hardware User Manual Book. 5 Mei 2018.
- [9] O. Boards, S. U. Units, P. Output, and S. Communications, "Machine Automation Controller NX1P2-."
- [10] Guy A. Boy. The Handbook of Human Machine Interaction. Florida Institut of Technology, USA.
- [11] Omron. NB Series Starting Guide Manual Book. 14 April 2018
- [12] A. Polyakov *et al*,"Relay Control Design using Attractive Ellipsoids Method To cite this version," no. Cdc, pp. 6646-6651, 2017.
- [13] Benjamin C. Groen. Investigation of DC Motors for Electric and Hybric Electric Motor Vehicle Applications Using an Infinitely Variable Transmission. Brigham Young University, April 2011.

- [14] Sarthak Chakraborty , Wireless Servo Motor Control using RF and MCU, Meghnad Saha Institut of Technology (Electrical Engineering 3rd year)
- [15] J . T. Wintermute and S. G. Hayler, " Transistor -Oscillator Limit Switch for," no. November, pp 518 - 519, 1960.
- [16] Rai Muhammad Amir, Faqir Muhammad Anjum, Moazam Rafiq Khan, Muhammad Imran. Application of Fourier Transport Infrared (FTIR) Spectroscopy. University Of Agriculture, Faisalabad Pakistan.

LAMPIRAN

Lampiran 1. Listing Program

```
TriggerEN:=TRUE;
CASE SeqIndex OF
  0:
    IF Trigger THEN

      SeqIndex:=1;

    END_IF;
  1:
    IF InputBarcode.ResultAvailable THEN
      GoodRead:=FALSE;
      BadRead:=FALSE;
      AckResultBarcode:=TRUE;
      SeqIndex:=2;
    ELSE
      SeqIndex:=10;
    END_IF;
  2:
    IF NOT InputBarcode.ResultAvailable THEN
      AckResultBarcode:=FALSE;
      SeqIndex:=10;
    END_IF;
  10:

    TriggerBarcode:=TRUE;
    IF InputBarcode.TriggerAck THEN
      DataTimer(IN:=FALSE);
      TriggerBarcode:=FALSE;
      SeqIndex:=20;
    ELSE
      DataTimer(IN:=TRUE,PT:=T#1000ms);
      IF DataTimer.Q THEN
        DataTimer(IN:=FALSE);
        SeqIndex:=1000;
      END_IF;
```

```

        END_IF;
20:    IF InputBarcode.ResultAvailable THEN
        DataTimer(IN:=FALSE);
        SeqIndex:=30;
    ELSE
        DataTimer(IN:=TRUE,PT:=T#500ms);
        IF DataTimer.Q THEN
            DataTimer(IN:=FALSE);
            SeqIndex:=1000;
        END_IF;
    END_IF;
30:    IF InputBarcode.ResultCode_DataRead THEN
        SeqIndex:=100;
    ELSIF NOT InputBarcode.ResultCode_DataRead
THEN
        SeqIndex:=110;
    END_IF;
100:    GoodRead:=TRUE;
    BadRead:=FALSE;

    Data:=AryToString(InputBarcode.ResultData[0],InputBarcode.ResultDataLength);
    SeqIndex:=200;
110:    GoodRead:=FALSE;
    BadRead:=TRUE;
    Data:='Not Read';
    SeqIndex:=200;
200:    Done:=TRUE;
    DataTimer(IN:=TRUE,PT:=T#500ms);
    IF DataTimer.Q THEN
        Done:=FALSE;
        SeqIndex:=210;
    END_IF;
210:

```

```

AckResultBarcode:=TRUE;
IF NOT InputBarcode.ResultAvailable THEN
    AckResultBarcode:=FALSE;
    SeqIndex:=1;

END_IF;

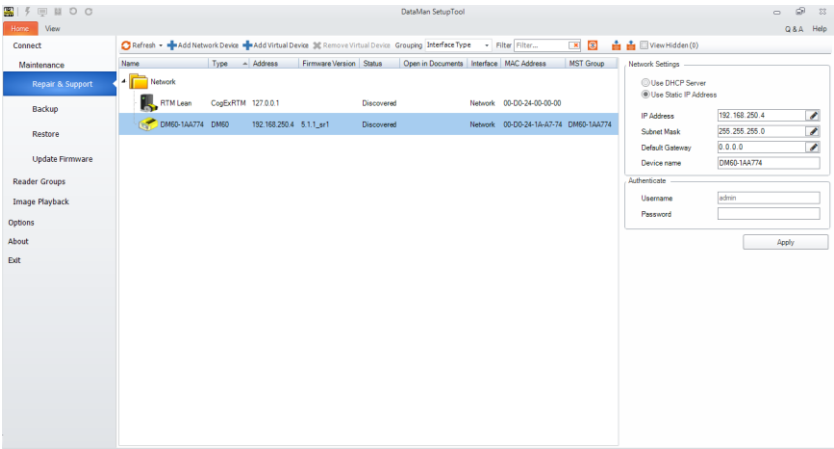
1000:
Error_BarcodeNotResponding:=TRUE;
IF NOT Trigger THEN
    Error_BarcodeNotResponding:=FALSE;
    TriggerBarcode:=FALSE;
    SeqIndex:=0;
END_IF;

END_CASE;

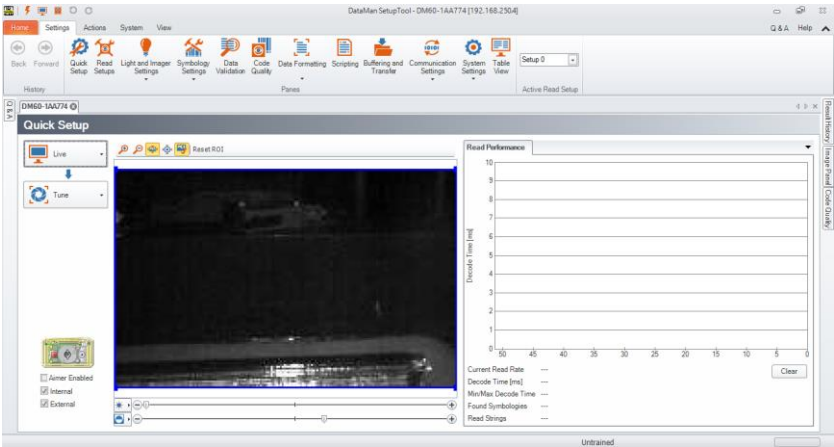
```

Lampiran 2.

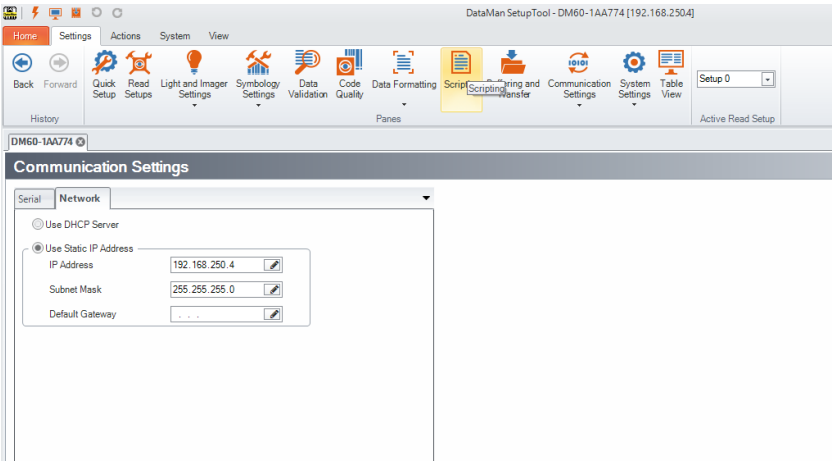
Tampilan awal *Dataman Software*.



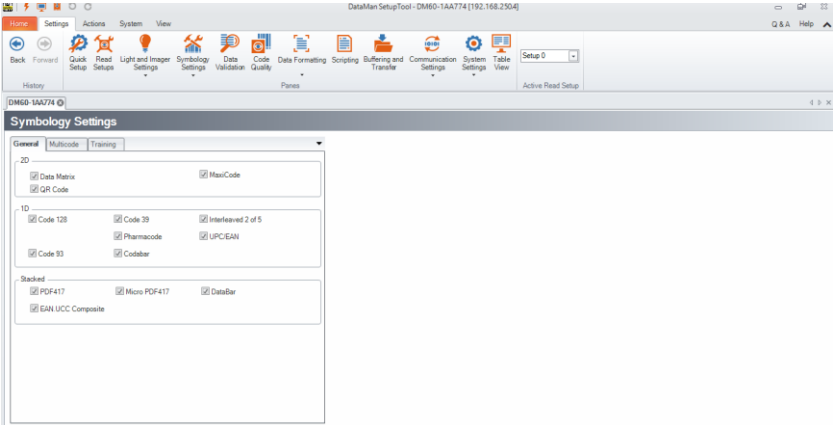
Tampilan Menu *Quick Setup*



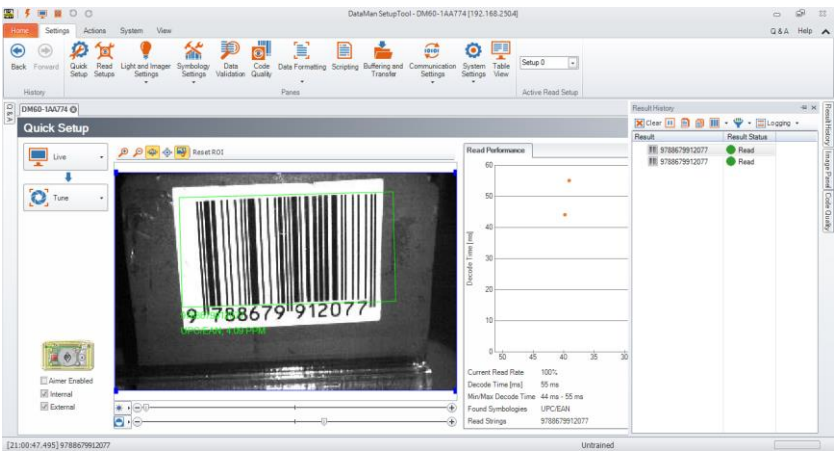
Tampilan Menu *Communication* pada *Datawam Software*



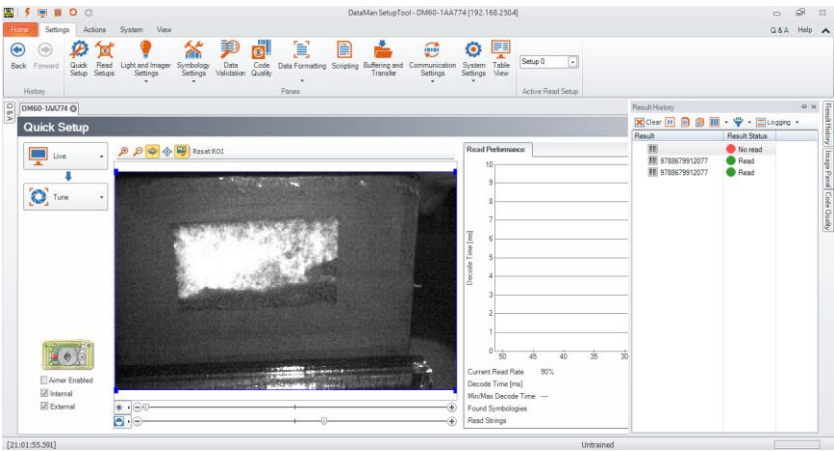
Tampilan Menu *Symbology Setting*



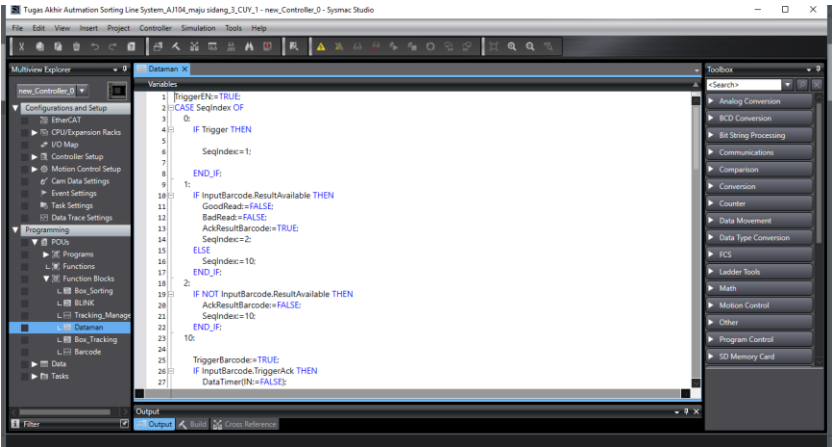
Hasil pengambilan gambar jika terbaca



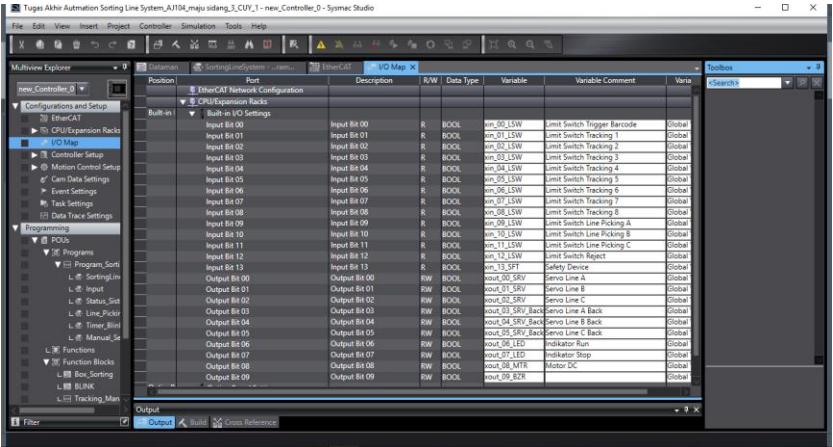
Hasil pengambilan gambar jika tidak terbaca



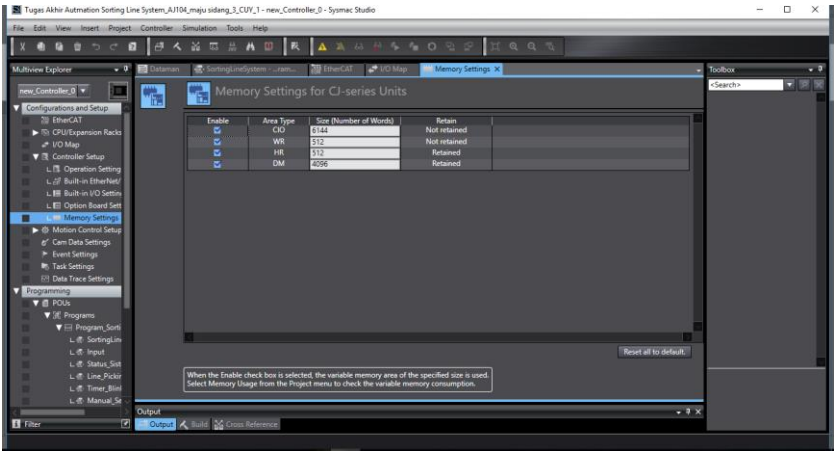
Tampilan Program dengan *Structure text* pada Sysmac Studio



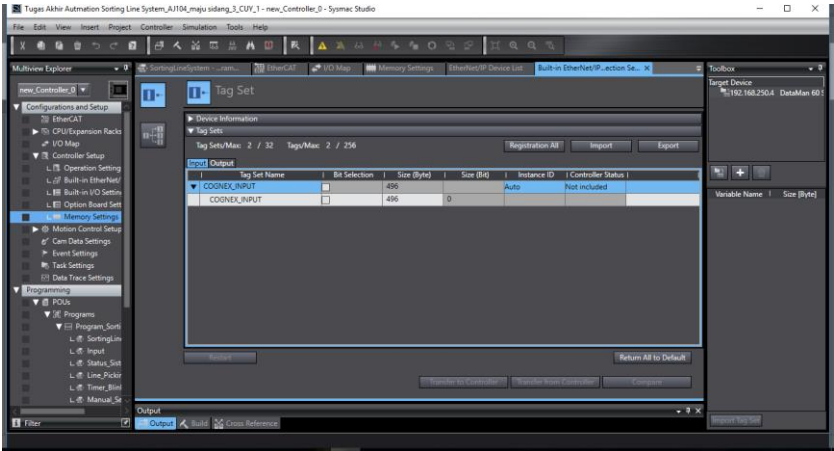
Tampilan men I/O Map pada Sysmac Studio



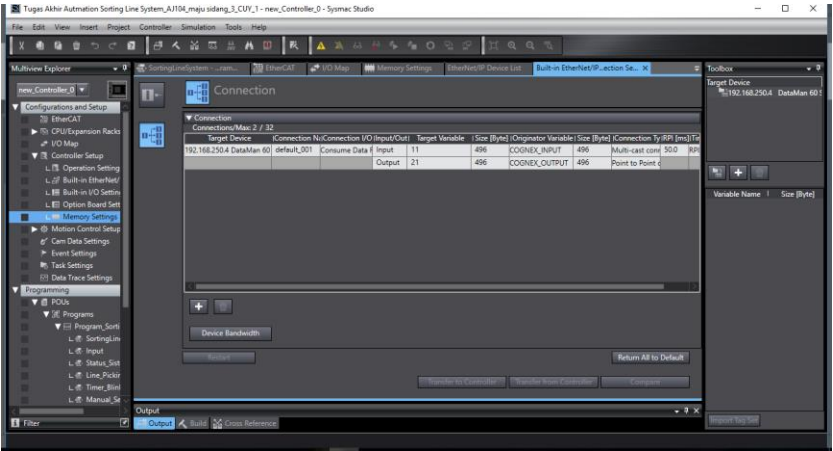
Tampilan menu *Memory Setting* pada Sysmac Studio



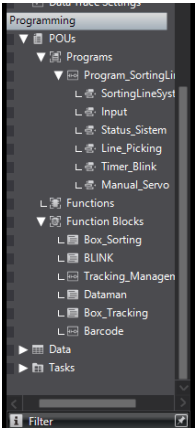
Tampilan menu *Tag set* pada Sysmac Studio



Tampilan menu *Connection* pada Symac Studio



POU (Program Organization Unit) pada Symac Studio



Lampiran 3. Datasheet DM60

Getting Started

About DataMan 60



DataMan 60 is a compact fixed-mount ID reader that among others offers the following advanced features:

- Smallest high-performance fixed mount reader
- Highest read rates for 1-D and 2-D codes
- Reliability and image feedback

DataMan 60 readers support RS-232, USB, and Ethernet interface connections plus discrete I/O.





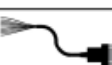


The DataMan 60 reader is packaged in an IP40-rated housing.

This document provides basic information about how to configure and use DataMan 60 readers. Additional information is available through the Windows Start menu or the DataMan Setup Tool Help menu after you install the DataMan software on your PC:

- The **DataMan Communications & Programming Guide** shows how to integrate your DataMan reader with your automation software and factory network.
Cognex->DataMan Software v x.x.x->Documentation->Communications And Programming
- The **DataMan Reader Configuration Codes** document provides printable 2-D codes that you can use to configure the DataMan reader.
Cognex->DataMan Software v x.x.x->Documentation->English->Reader Configuration Codes
- The **DataMan Fixed Mount Readers Reference** is a complete online hardware reference for the DataMan fixed-mount ID readers.
Cognex->DataMan Software v x.x.x->Documentation->English->DM60 ->Fixed Mount Reference Manual
- The **DataMan Questions and Answers** document provides context-sensitive information. You can view this help inside the DataMan Setup Tool or as a stand-alone help file.
Cognex->DataMan Software v x.x.x->Documentation->DM60->Questions and Answers
- The **Release Notes** list detailed system requirements and additional information about this DataMan software release.
Cognex->DataMan Software v x.x.x->Documentation->DataMan v x.x.x Release Notes

DataMan 60 Accessories

Cables

USB Cable, 1.5 m (DM100-USB-000)	
USB Cable, 3 m (DM100-USB-030)	
USB and Flying Leads I/O Cable, 2.0 m (DM-USBIO-00)	
RS-232 and Flying Leads I/O Cable, 2.5 m (DM-RS232IO-00)	
RS-232 Cable, 1.5 m (DM100-RS232-000)	
Extension Cable, 5 m (DM100-EXTCBL-000)	
Flying Leads Connection Cable, 5 m (DM50-PWRIO-05)	
Ethernet Cable: use any standard CAT5/5e, SF/FTP or S/FTP cable	
RS-232/USB adapter connector (DM100-PATCH-000)	

Power Supply

Power Supply, 6 V (DM100-PWR-000)	
-----------------------------------	---

Mounting Brackets

Pivot Mounting Bracket (DM100-PIVOTM-00)	
--	--

Universal Mounting Bracket (DM50-UBRK-000)




Control Box

Control Box (DM-CTRLBOX-00)



DataMan 60 Systems

	Omni-directional 1-D Code Reading	IDQuick™ — Well marked 2-D Code Reading	1DMax+™ with HOTBARS™
DataMan 60L (DMR-60L-00)			√
DataMan 60QL (DMR-60QL-00)	√		√
DataMan 60S (DMR-60S-00)	√	√	√


Model Limitations

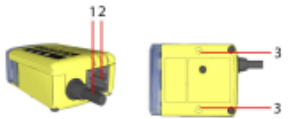
S models do not have Burst and Continuous trigger modes.

	60L	60QL	60S
1-D and Stacked Codes	Yes, oriented	Yes, omnidirectional	Yes, omnidirectional
Omnidirectional 1-D Codes	No	Yes	Yes
2-D Codes	No	No	Yes
Algorithm	1DMax, Hotbars	1DMax, Hotbars	1DMax, Hotbars, IDQuick
Acquisition	Max 60 fps	Max 60 fps	Max 2 fps
Max Decode Rate	45/sec	45/sec	2/sec
Trigger	Manual; External: single, burst and continuous; Internal: self and presentation	Manual; External: single, burst and continuous; Internal: self and presentation	Manual; External: single Internal: self and presentation

Reader Layout

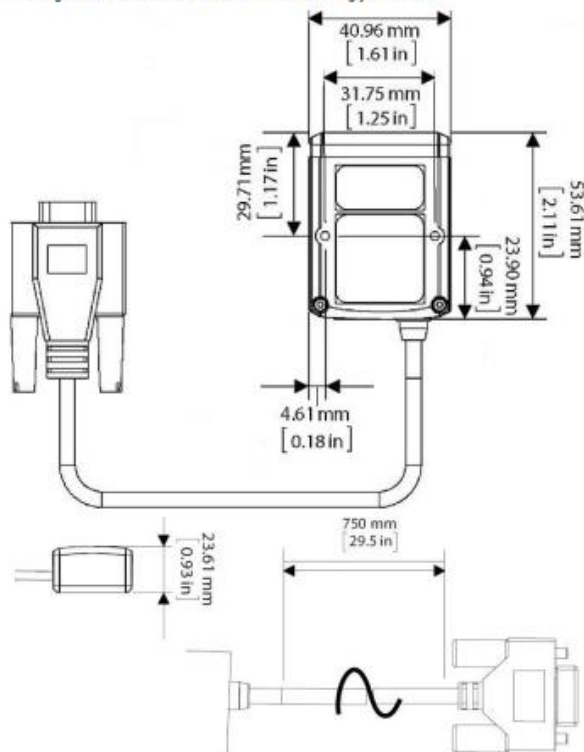
The following image shows the characteristics of the DataMan 60reader.

	
1	Internal illumination
2	3-position lens cap
3	Focal position indicator
4	External illumination connector
5	Status LEDs: <ul style="list-style-type: none"> • Power: YELLOW = Power ON • Communication: Flashing YELLOW • Good / bad read: GREEN = good read / RED = bad read • Error: RED = error, check device log
6	LED aimer
7	Lens

	
1	Directly connected cable terminating in a DB15 connector providing: power, IO, USB, and RS-232 connectivity
2	Ethernet RJ-45 connector
3	Mounting points

Dimensions

Observe the following DataMan 60 reader dimensions when installing your reader.



The attachment points have the following characteristics:

- M3 thread
- Maximum thread depth: 6 mm
- 1.3 N-m (11.5 in-lb) maximum torque

CAUTION: Use both attachment points when mounting your device.

Setting Up Your DataMan 60

Setting the Focus Position

DataMan 60 can operate in one of three distance ranges.

To set the focus position:

1. Remove the lens cover.



2. Set the focus position to 45, 70, or 110: turn the lens cap clockwise (45->70->110) to focus to a larger distance; turn the lens cap counter-clockwise (110->70->45) to focus to a shorter distance.
3. Reattach the front cover.

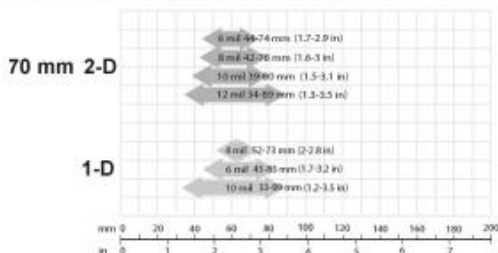


Field of View and Reading Distances

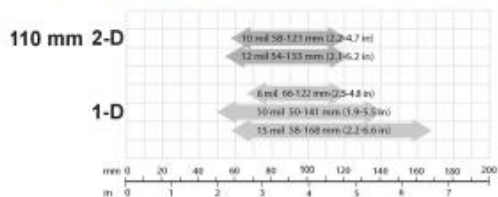
The following charts show the field of view (FoV) of the DataMan 60 series readers. Reading distance values are also provided for 1-D and 2-D example code distances.

This chart shows the supported range of reading distances at **45 mm** focus position.

This chart shows the supported range of reading distances at **70 mm** focus position.



This chart shows the supported range of reading distances at **110 mm** focus position.



The following table shows the FoV widths in mm at various distances.

Distances in mm	DataMan 60
45	36
70	56
110	80

DataMan 60 Imager Specifications

Specification	DataMan 60 Series Imager
Image Sensor	1/3 inch CMOS
Image Sensor Properties	4.51 mm x 2.88 mm (H x V), 6.0 μ m square pixels
Image Resolution (pixels)	752 x 480
Electronic Shutter Speed	18 μ s to 25 ms exposure
Image Acquisition	up to 60 fps at full resolution
Lens Type	6.2 mm, F:5.3 focal position M12 lens with IR blocking filter

DataMan 60 Specifications

Weight	98 g (including cable)		
Operating Temperature	0 °C — 40 °C (32 °F — 104 °F)		
Storage Temperature	-10 °C — 60 °C (-14 °F — 140 °F)		
Maximum Humidity	95% (non-condensing)		
Environmental	IP40		
LED Safety	IEC 62471: Exempt risk group, no further labeling is required.		
Codes	1-D barcodes: Codabar, Code 39, Code 128, and Code 93, Interleaved 2 of 5, Pharma, Postal, UPC/EAN/JAN 2-D barcodes: Data Matrix™ QR Code and microQR Code, MaxiCode, RSSCS, PDF 417, MicroPDF 417		
Discrete I/O Operating Limits	Output 0,1	I_{MAX} @ 24 VDC	25 mA
		V_{MAX}	24 V
	Output 2	Source V_{TYP}	4 V
		Sink V_{IH}	4 V - V_{PSU}
		V_{IL}	0 — 2 V
	Input 0 (Trigger)	V_{IH}	4 — 26 V
	Input 1	V_{IL}	0 — 2 V
		I_{TYP}	3 mA
Power Supply Requirements	V_{PSU} 4.5 — 24 VDC 2.5 W maximum LPS or NEC class 2 power supply		
Ethernet Speed	10/100		
Duplex Mode	Full duplex or half duplex		

Using Your DataMan 60

Install Your DataMan 60

Perform the following steps to install the DataMan Setup Tool:

1. Check the **DataMan Release Notes** for a full list of system requirements.
2. Download the DataMan Setup Tool from <http://www.cognex.com/support/dataman> and follow the on-screen steps.
3. Connect the DataMan 60 reader to your PC.
4. Choose **Start > Programs > Cognex > DataMan Software v x.x.x > Setup Tool** to launch the DataMan Setup Tool. The detected readers will appear under the **COM ports**.
5. Click **Refresh** to update the list of connected devices.
6. Select a COM port that lists DataMan 60 and click **Connect**.

Follow the steps below to connect your reader to power and network:

1. Connect the cable on the back of the device to either a USB adapter cable with power tap or to an RS-232 adapter cable with power tap.
2. Connect a 6 V power supply.
3. Connect the reader to an Ethernet network.

DataMan 60 Triggering

DataMan 60 readers support the following trigger modes:

- **Self:** At an interval you configure, the reader automatically detects and decodes codes in its field of view. If you set a higher re-read delay than the trigger interval, there is a code output only once until the code is out of the field of view for the duration of the re-read delay.
- **Single (external trigger):** Acquires a single image and attempts to decode any symbol it contains, or more than one symbol in cases where multicode is enabled. The reader relies on an external trigger source.
- **Presentation:** Scans, decodes and reports a single code in the field of view. The reader relies on an internal timing mechanism to acquire images.
- **Manual:** Begins acquiring images when you press the trigger button on the reader, and continues acquiring images until a symbol is found and decoded or you release the button.
- **Burst:** Performs multiple image acquisitions based on an external trigger and decodes any symbol appearing in a single image or within a sequence of images, or multiple symbols in a single image or within a sequence of images when multicode is enabled. You can control the number of images within each burst and the interval between image acquisitions.
- **Continuous:** Begins acquiring images based on a single external trigger and continues to acquire and decode images until a symbol is found and decoded, or until multiple images containing as many codes as specified in multicode mode are located, or until the trigger is released. You can configure your reader to acquire images based on the start and stop signal from separate digital IO pulses.

S models do not have Burst and Continuous trigger modes.

External Triggers

If you are using external triggering, you can use any of these methods to trigger your DataMan 60 series reader:

- Send a pulse on the I/O cable:
 - Input 0 (white)
 - Input 1 (white/black)
- Send a serial trigger command over the RS-232 connection or the Ethernet connection.
- Press <CTRL>->T on the keyboard while the DataMan Setup Tool has the input focus.
- Click the Trigger button in the DataMan Setup Tool:

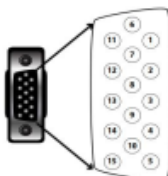


Connections, Optics and Lighting

I/O Cable

The I/O cable provides access to trigger and high-speed outputs. Unused wires can be clipped short or tied back using a tie made of non-conductive material. For RS-232, use the Power Supply return path for ground.

Note: GND (Pin 4) is connected to the reader housing, cable shield, and DB15 shell.

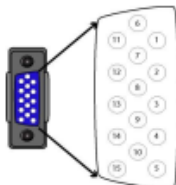


This is a male connector/plug.

PIN	Color	Signal
1	Brown	Reserved
2	Green	TxD
3	Green/Black	RxD
4	Red and Brown/White	GND
5	Red/Black	DC+ (system power, 5-24 VDC)
6	Blue	RTS
7	Blue/White	Output-0
8	White	Input-0
9	White/Black	Input-1
10	Light Blue	CTS
11	Light Blue/Black	Output-1
12	Light Blue/Yellow	Output-Common
13	Light Blue/Green	Output-Strobe
14	Yellow	Reserved
15	Yellow/Black	Reserved

USB and Flying Leads I/O Cable

You can connect a cable with USB and flying leads (DM-USBIO-00) to the cable that is attached to the device. The following table shows the pinout and color description of the flying leads.

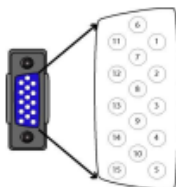


This is a female connector/socket.

PIN	Color	Signal
4	Black	GND
7	Blue/White	Output-0
8	White	Input-0
9	White/Black	Input-1
11	Light Blue/Black	Output-1
12	Light Blue/Yellow	Output-Common
13	Light Blue/Green	Output-Strobe

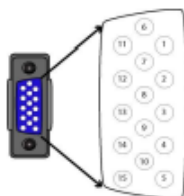
RS-232 and Flying Leads I/O Cable

You can connect a cable with RS-232 and flying leads (DM-RS232IO-00) to the cable that is attached to the device. The following table shows the pinout and color description of the flying leads.



This is a female connector/socket.

PIN	Color	Signal
4	Black	GND
5	Brown/White	VDC
7	Blue/White	Output-0

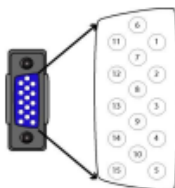


This is a female connector/socket.

PIN	Color	Signal
8	White	Input-0
9	White/Black	Input-1
11	Light Blue/Black	Output-1
12	Light Blue/Yellow	Output-Common
13	Light Blue/Green	Output-Strobe

Flying Leads Cable

You can connect a cable with flying leads (DM50-PWRIO-05) to the cable that is attached to the device. The following table shows the pinout and color description of the flying leads.

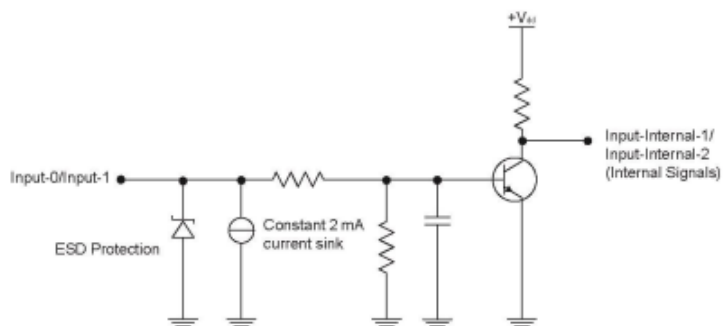


This is a female connector/socket.

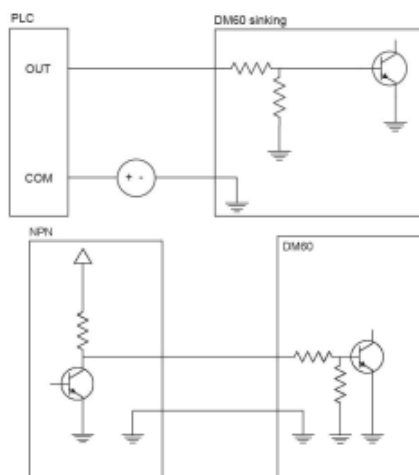
IN	Color	Signal
2	Green	TxD
3	Green/Black	RxD
4	Black	GND
5	Brown/White	DC+ (system power, 5-24 VDC)
6	Blue	RTS
7	Blue/White	Output-0
8	White	Input-0
9	White/Black	Input-1
10	Light Blue	CTS
11	Light Blue/Black	Output-1
12	Light Blue/Yellow	Output-Common
13	Light Blue/Green	Output-Strobe

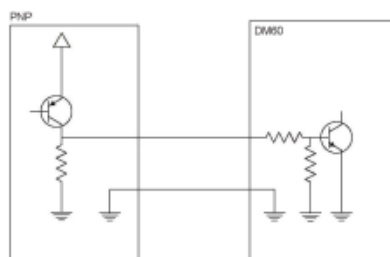
Digital Input Lines

Inputs are not galvanic isolated but need to be referenced to ground.



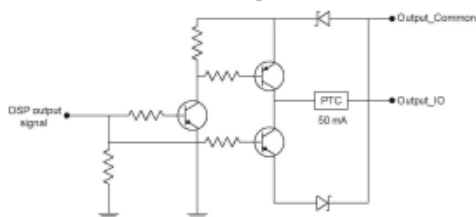
External Wiring Examples: Digital Input Lines





Digital Output Lines

The digital outputs can be used as either NPN (pull-down) or PNP (pull-up) lines. For NPN lines, the external load should be connected between the output and the positive supply voltage (<24 V). The outputs pull down to less than 3 V when ON, which causes current to flow through the load. When the outputs are OFF, no current flows through the load. Outputs are not galvanic isolated but need to be referenced to ground.



NPN (pull down) output type characteristics

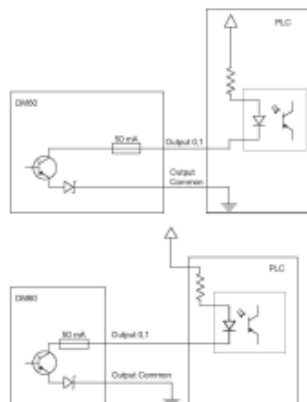
Applied voltage	24 VDC or less
Residual voltage	0.85V or less
Maximum sink current	25 mA
Short-circuit current	100 mA or less
Short-circuit protection	multifuse - 50 mA

PNP (pull up) output type characteristics

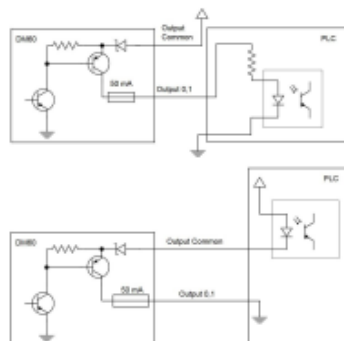
Applied voltage	24 V or less
Residual voltage	0.8 V or less
Maximum sink current	25 mA
Short-circuit current	50 mA or less
Short-circuit protection	multifuse - 50 mA

External Wiring Examples: Digital Output Lines

NPN:



PNP:



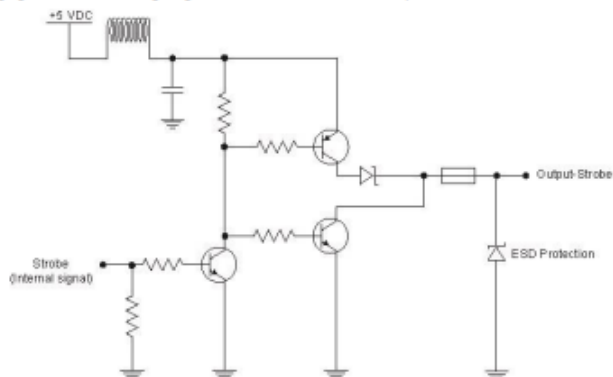
Illumination Strobe Output

The strobe output is provided by a diode that is added to the push-pull circuit, with series to the pull-up transistor. This diode blocks the higher voltage when the output is pulled up when used as open-collector type driving, but enables the driving of high level in TTL mode.

TTL output type characteristics	
High level	4.0-5.0 V
Low level	0-0.4 V
Output current	25 mA
Short-circuit current	125 mA
Short-circuit protection	multifuse - 50 mA

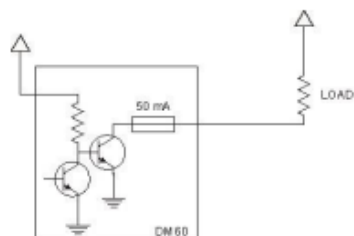
Open-collector output type characteristics	
Output voltage range	0-24 V
Low level	0-0.4 V
Output current	25 mA max
Short-circuit current	125 mA
Short-circuit protection	multifuse - 50 mA

The following figure shows the wiring diagram of the illumination strobe output:

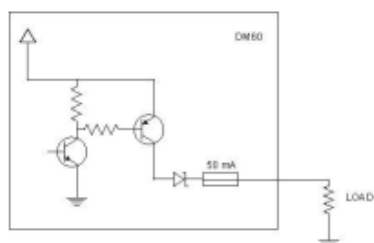


External Load Strobe Output

NPN:



PNP:



Multi-Port Connections

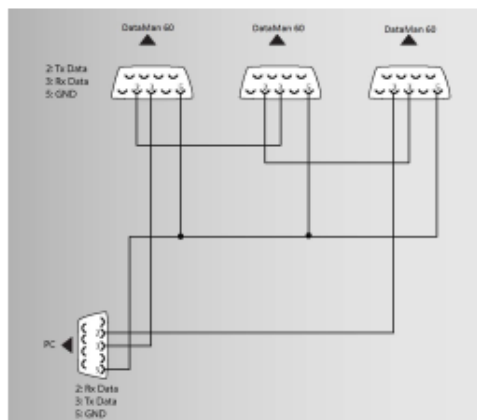
You can connect multiple DataMan 60 readers to a single PC (or other device equipped with a serial port) using a multi-port connection.

A multi-port connection creates a daisy-chain of readers. Each reader receives serial data from the previous reader and transmits it to the next reader. When a reader transmits data, it is passed through each of the readers in the chain between it and the PC.

Because of the large number of possible configurations, Cognex does not supply cabling for multi-port DataMan 60 connections. Instead, you must construct your own cable that meets the requirements of your system configuration.

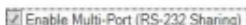
The cable must provide a DB9 connector for each DataMan 60 serial cable and a DB9 connector for the PC serial port. Each DB9 connector must provide Tx Data, Rx Data, and ground. The Tx Data and Rx Data pins on adjacent connectors must be connected to provide the multi-port connection.

The following diagram shows how to create a multi-port cable for a 3-reader system:



Configuring for Multi-Port Operation

You must connect the DataMan Setup Tool to each DataMan 60 in turn and enable multi-port operation. Click the **Communication Settings** pane, the **Serial** tab and check the **Enable Multi-Port (RS-232 Sharing)** checkbox.



There is no guaranteed delivery order when multiple readers transmit data using a multi-port connection; read results may arrive at the PC in any order. You can configure each DataMan 60 reader in a multi-port connection to add identifying data to each read result. Your PC application can then determine which reader produced a specific read result.

To do this, click on the **Data Formatting** pane, go to **Standard Formatting**, select the **Data Matrix** tab and enter text in the **Leading Text** field. (You can also add trailing text by entering text in the **Terminating Text** field.)

Standard Formatting

Data Matrix

QR Code / MaxiCode / Aztec Code

DotCode

1D / Stacked / Postal

Universal

Leading Text

Reader1

Data

General

Validation

<Sub-string>

<Full string>

Multi-Port Usage Notes

You can obtain the best results when using multi-port connections by keeping the following usage guidelines in mind as you design your system:

- The maximum cable length between any two DataMan readers or between the PC and any DataMan reader should be no greater than 15 meters.
- There is no fixed limit to the number of DataMan readers that you can connect to a single PC. Each reader introduces a delay of about 100 msec when it retransmits received serial data. If you have 5 readers, this means that there will be a 400 msec delay between the time the first reader in the chain transmits data and the PC receives it.
- Each DataMan reader must receive a hardware trigger signal on its Input 0 line. You can wire the input ports to a common trigger signal or you can provide individual triggers for each reader.
- If any reader in the multi-port chain loses power or becomes disconnected, then no data from any other reader will be transmitted.
- If a DataMan is transmitting its own read result, it will buffer any data received from another reader until it has finished its own data transmission. If a DataMan reader is transmitting another reader's data, it will buffer its own data if it receives a trigger signal while it is processing the other reader's data.
- If you use a single power supply for multiple readers, make sure that the power supply can provide enough power for all of the readers.
- You cannot connect a reader to the DataMan Setup Tool over RS-232 once multiport is enabled. You must first scan the Disable Multi-Port code from the *Reader Configuration Codes*, available from the Start menu.

Lampiran 4. EDS Cognex DM60

\$ EZ-EDS Version 3.9 Generated Electronic
Data Sheet

[File]

```
DescText = "DataMan 60 Series
Reader";
CreateDate = 04-15-2013;
CreateTime = 09:00:00;
ModDate = 09-03-2015;
ModTime = 15:35:30;
Revision = 1.2;
HomeURL = "http://www.cognex.com/";
```

[Device]

```
VendCode = 678;
VendName = "Cognex";
ProdType = 773;
ProdTypeStr = "DataMan 60 Series
Reader";
ProdCode = 1295;
MajRev = 1;
MinRev = 5;
ProdName = "DataMan 60 Series
Reader";
Catalog = "DMR-60X-00";
Icon = "DM60.ico";
```

[Device Classification]

```
Class1 = EtherNetIP;
```

[Params]

```
Param2 =
    0,
    '',
    0x0000,
    0xC7,
```

```

        2,
        "Input Size",
        "Bytes",
        "",
        2,500,496,
        ', ',
        ', ',
        ;
Param3 =
    0,
    ', ',
    0x0000,
    0xC7,
    2,
    "Output Size",
    "Bytes",
    "",
    2,500,496,
    ', ',
    ', ',
    ;

Param4 =
    0,
    ', ',
    0x0000,
    0xC1,
    1,
    "Trigger Enable",
    "",
    "",
    0,1,0,
    ', ',
    ', ',
    ;

Param5 =
    0,
    ', '

```

```

        0x0000,
        0xC1,
        1,
        "Trigger",
        "",
        "",
        0,1,0,
        ',,,'
        ',,,'
        ;
Param6 =
        0,
        ',,
        0x0000,
        0xC1,
        1,
        "Buffer Results Enable",
        "",
        "",
        0,1,0,
        ',,,'
        ',,,'
        ;
Param7 =
        0,
        ',,
        0x0000,
        0xC1,
        1,
        "Results Ack",
        "",
        "",
        0,1,0,
        ',,,'
        ',,,'
        ;
Param9 =
        0,

```

```

        ',
        0x0000,
        0xC1,
        1,
        "Soft Event 0",
        "",
        "",
        0,1,0,
        ',,
        ',,
        ;
Param10 =
        0,
        ',
        0x0000,
        0xC1,
        1,
        "Soft Event 1",
        "",
        "",
        0,1,0,
        ',,
        ',,
        ;
Param11 =
        0,
        ',
        0x0000,
        0xC1,
        1,
        "Soft Event 2",
        "",
        "",
        0,1,0,
        ',,
        ',,
        ;
Param12 =

```

```

0,
'',
0x0000,
0xC1,
1,
"Soft Event 3",
'',
'',
0,1,0,
'',
'',
;
Param13 =
0,
'',
0x0000,
0xC1,
1,
"Soft Event 4",
'',
'',
0,1,0,
'',
'',
;
Param14 =
0,
'',
0x0000,
0xC1,
1,
"Soft Event 5",
'',
'',
0,1,0,
'',
'',
;

```

```

Param15 =
    0,
    ,,
    0x0000,
    0xC1,
    1,
    "Soft Event 6",
    "",
    "",
    0,1,0,
    ,,,,
    ,,,,
    ;

Param16 =
    0,
    ,,
    0x0000,
    0xC1,
    1,
    "Soft Event 7",
    "",
    "",
    0,1,0,
    ,,,,
    ,,,,
    ;

Param17 =
    0,
    ,,
    0x0000,
    0xC7,
    2,
    "User Data Option",
    "",
    "",
    0,65535,0,
    ,,,,
    ,,,,

```

```

;
Param18 =
0,
'',
0x0000,
0xC7,
2,
"User Data Length",
'',
'',
0,65535,0,
'',
'',
;
Param19 =
0,
'',
0x0000,
0xD1,
1,
"User Data",
'',
'',
0,255,0,
'',
'',
;
Param22 =
0,
'',
0x0000,
0xC1,
1,
"Trigger Ready",
'',
'',
0,1,0,

```



```

        ', ',
        ', ',
        ;
Param23 =
    0,
    ',
    0x0000,
    0xC1,
    1,
    "Trigger Ack",
    "",
    "",
    0,1,0,
    ', ',
    ', ',
    ;
Param24 =
    0,
    ',
    0x0000,
    0xC1,
    1,
    "Acquiring",
    "",
    "",
    0,1,0,
    ', ',
    ', ',
    ;
Param25 =
    0,
    ',
    0x0000,
    0xC1,
    1,
    "Missed Acq",
    "",
    "",

```

```

        0,1,0,
        ',',' ',
        ',',' ',
        ;
Param26 =
        0,
        ',
        0x0000,
        0xC1,
        1,
        "Decoding",
        '"',
        '"',
        0,1,0,
        ',',' ',
        ',',' ',
        ;
Param27 =
        0,
        ',
        0x0000,
        0xC1,
        1,
        "Decode Complete",
        '"',
        '"',
        0,1,0,
        ',',' ',
        ',',' ',
        ;
Param28 =
        0,
        ',
        0x0000,
        0xC1,
        1,
        "Results Buffer Overrun",
        '"',

```

```

        "",
        0,1,0,
        ",",
        ",",
        ;
Param29 =
        0,
        "",
        0x0000,
        0xC1,
        1,
        "Results Available",
        "",
        "",
        0,1,0,
        ",",
        ",",
        ;
Param30 =
        0,
        "",
        0x0000,
        0xC1,
        1,
        "General Fault",
        "",
        "",
        0,1,0,
        ",",
        ",",
        ;
Param31 =
        0,
        "",
        0x0000,
        0xC1,
        1,
        "Soft Event Ack 0",

```

```

        "" ,
        "" ,
        0,1,0,
        "" ,
        "" ,
        ;
Param32 =
        0,
        ,
        0x0000,
        0xC1,
        1,
        "Soft Event Ack 1",
        "" ,
        "" ,
        0,1,0,
        "" ,
        "" ,
        ;
Param33 =
        0,
        ,
        0x0000,
        0xC1,
        1,
        "Soft Event Ack 2",
        "" ,
        "" ,
        0,1,0,
        "" ,
        "" ,
        ;
Param34 =
        0,
        ,
        0x0000,
        0xC1,
        1,

```

```

        "Soft Event Ack 3",
        "",
        "",
        0,1,0,
        ",",
        ",",
        ;
Param35 =
    0,
    "",
    0x0000,
    0xC1,
    1,
    "Soft Event Ack 4",
    "",
    "",
    0,1,0,
    ",",
    ",",
    ;
Param36 =
    0,
    "",
    0x0000,
    0xC1,
    1,
    "Soft Event Ack 5",
    "",
    "",
    0,1,0,
    ",",
    ",",
    ;
Param37 =
    0,
    "",
    0x0000,
    0xC1,

```

```

1,
"Soft Event Ack 6",
"",
"",
0,1,0,
,,,,
,,,,
;
Param38 =
0,
'',
0x0000,
0xC1,
1,
"Soft Event Ack 7",
"",
"",
0,1,0,
,,,,
,,,,
;
Param39 =
0,
'',
0x0000,
0xC7,
2,
"Trigger ID",
"",
"",
0,65535,0,
,,,,
,,,,
;
Param40 =
0,
'',
0x0000,

```

```

        0xC7,
        2,
        "Result ID",
        "",
        "",
        0,65535,0,
        ',,,'
        ',,,'
        ;
Param41 =
        0,
        ',,
        0x0000,
        0xC7,
        2,
        "Result Code",
        "",
        "",
        0,65535,0,
        ',,,'
        ',,,'
        ;
Param42 =
        0,
        ',,
        0x0000,
        0xC7,
        2,
        "Result Extended",
        "",
        "",
        0,65535,0,
        ',,,'
        ',,,'
        ;
Param43 =
        0,
        ',,

```

```

        0x0000,
        0xC7,
        2,
        "Result Length",
        "",
        "",
        0,65535,0,
        ',,,'
        ',,,'
        ;
Param44 =
        0,
        ',,
        0x0000,
        0xD1,
        1,
        "Result Data",
        "",
        "",
        0,255,0,
        ',,,'
        ',,,'
        ;

[Assembly]
        Assem11 =
            "input format",
            ,
            ,
            0x0000,
            ',,
Trigger Ready      1,Param22,          $
Trigger Ack        1,Param23,          $
Acquiring          1,Param24,          $

```


	1, Param25,	\$
Missed Acq		
	4,,	\$
Reserved		
	1, Param26,	\$
Decoding		
	1, Param27,	\$
Decode Complete		
	1, Param28,	\$
Results Buffer Overrun		
	1, Param29,	\$
Results Available		
	3,,	\$
Reserved		
	1, Param30,	\$
General Fault		
	1, Param31,	\$
Soft Event Ack 0		
	1, Param32,	\$
Soft Event Ack 1		
	1, Param33,	\$
Soft Event Ack 2		
	1, Param34,	\$
Soft Event Ack 3		
	1, Param35,	\$
Soft Event Ack 4		
	1, Param36,	\$
Soft Event Ack 5		
	1, Param37,	\$
Soft Event Ack 6		
	1, Param38,	\$
Soft Event Ack 7		
	24,,	\$
Reserved		

Trigger ID	16,Param39,	\$
Result ID	16,Param40,	\$
Result Code	16,Param41,	\$
Result Extended	16,Param42,	\$
Result Length	16,Param43,	\$
Result Data	3872,Param44;	\$
Assem21 = "output format", , , 0x0000, '', 1,Param4,		
Trigger Enable	1,Param5,	\$
Trigger	1,Param6,	\$
Buffer Results Enable	1,Param7,	\$
Results Ack	4,,	\$
Reserved		
Soft Event 0	1,Param9,	\$
Soft Event 1	1,Param10,	\$

	1,Param11,	\$
Soft Event 2		
	1,Param12,	\$
Soft Event 3		
	1,Param13,	\$
Soft Event 4		
	1,Param14,	\$
Soft Event 5		
	1,Param15,	\$
Soft Event 6		
	1,Param16,	\$
Soft Event 7		
	16,,	\$
Reserved		
	16,Param17,	\$
User Data Option		
	16,Param18,	\$
User Data Length		
	3936,Param19;	\$
User Data		
[Connection Manager]		
Connection1 =		
0x04010002,		\$
trigger & transport		
		\$
		\$
value bits		
		\$
2: 0-15 = supported transport classes		
(class 1)		
		\$
1: 16 = cyclic (1 = supported)		

0:	17 = change of state (0 = not supported)	\$
0:	18 = on demand (0 = not supported)	\$
0:	19-23 = reserved (must be zero)	\$
4:	24-27 = exclusive owner	\$
0:	28-30 = reserved (must be zero)	\$
0:	31 = client 0 (don't care for classes 0 and 1)	\$
	0x44640405,	\$
	connection parameters	\$
		\$
	value bits	\$
1:	0 = O=>T fixed (1 = supported)	\$
0:	1 = O=>T variable (0 = not supported)	\$
1:	2 = T=>O fixed (1 = supported)	\$
0:	3 = T=>O variable (0 = not supported)	\$
0:	4-7 = reserved (must be zero)	\$
4:	8-11 = O=>T header (4 byte run/idle)	\$
0:	12-15 = T=>O header (no header)	\$
4:	16-19 = O=>T connection type (point-to-point)	\$

```

                                                                    $
6:  20-23 = T=>O connection type (multicast,
point-to-point)
                                                                    $
4:  24-27 = O=>T priority (scheduled)
                                                                    $
4:  28-31 = T=>O priority (scheduled)
                                                                    $
                                ,Param3,Assem21,
                                                                    $
O=>T RPI,Size,Format
                                ,Param2,Assem11,
                                                                    $
T=>O RPI,Size,Format
                                ',
                                                                    $
config part 1 (not used)
                                ',
                                                                    $
config part 2 (not used)
                                "Consume Data From/Produce
Data To",    $ connection name
                                "",
                                                                    $
Help string
                                "20 04 24 01 2C 15 2C 0B";

                                Connection2 =
                                0x02010002,
                                                                    $
trigger & transport
                                                                    $
                                                                    $
value  bits
                                                                    $
2:    0-15 = supported transport classes
(class 1)
                                                                    $
1:    16 = cyclic (1 = supported)
                                                                    $
0:    17 = change of state (0 = not
supported)
                                                                    $
0:    18 = on demand (0 = not supported)

```

0:	19-23 = reserved (must be zero)	\$
2:	24-27 = input only	\$
0:	28-30 = reserved (must be zero)	\$
0:	31 = client 0 (don't care for classes 0 and 1)	\$
	0x44640305,	\$
	connection parameters	\$
		\$
	value bits	\$
1:	0 = O=>T fixed (1 = supported)	\$
0:	1 = O=>T variable (0 = not supported)	\$
1:	2 = T=>O fixed (1 = supported)	\$
0:	3 = T=>O variable (0 = not supported)	\$
0:	4-7 = reserved (must be zero)	\$
3:	8-11 = O=>T header (0 byte heartbeat)	\$
0:	12-15 = T=>O header (no header)	\$
4:	16-19 = O=>T connection type (point-to-point)	\$
6:	20-23 = T=>O connection type (multicast, point-to-point)	\$
4:	24-27 = O=>T priority (scheduled)	\$

```

4:  28-31 = T=>O priority (scheduled)
                                     $
                                     $
                                     ,0,,
                                     $
O=>T RPI,Size,Format
                                     ,Param2,Assem11,
                                     $
T=>O RPI,Size,Format
                                     $
                                     ''
config part 1 (not used)
                                     $
                                     ''
config part 2 (not used)
                                     "Consume Data From",
                                     $
connection name
                                     "",
                                     $
Help string
                                     "20 04 24 01 2C FE 2C 0B";
$ inputs only path

        Connection3 =
                0x04010002,
trigger & transport
                                     $
                                     $
value  bits
                                     $
2:  0-15 = supported transport classes
(class 1)
                                     $
1:  16 = cyclic (1 = supported)
                                     $
0:  17 = change of state (0 = not
supported)
                                     $
0:  18 = on demand (0 = not supported)
                                     $
0:  19-23 = reserved (must be zero)
                                     $
4:  24-27 = exclusive owner

```

		\$
0:	28-30 = reserved (must be zero)	
		\$
0:	31 = client 0 (don't care for classes 0 and 1)	
		\$
	0x44443405,	\$
	connection parameters	
		\$
		\$
	value bits	
		\$
1:	0 = O=>T fixed (1 = supported)	
		\$
0:	1 = O=>T variable (0 = not supported)	
		\$
1:	2 = T=>O fixed (1 = supported)	
		\$
0:	3 = T=>O variable (0 = not supported)	
		\$
0:	4-7 = reserved (must be zero)	
		\$
4:	8-10 = O=>T header (4 byte run/idle)	
		\$
0:	11 = reserved (must be zero)	
		\$
3:	12-14 = T=>O header (0 byte heartbeat)	
		\$
0:	15 = reserved (must be zero)	
		\$
4:	16-19 = O=>T connection type (point-to-point)	
		\$
4:	20-23 = T=>O connection type (point-to-point)	
		\$
4:	24-27 = O=>T priority (scheduled)	


```

4:  28-31 = T=>O priority (scheduled)
                                     $
                                     $
                                     ,Param3,Assem21,
                                     $
O=>T RPI,Size,Format
                                     ,0,,
                                     $
T=>O RPI,Size,Format
                                     '
config part 1 (not used)
                                     '
config part 2 (not used)
                                     "Produce Data To",
connection name
                                     "",
Help string
                                     "20 04 24 01 2C 15 2C FE";
$ outputs only path

[Port]
    Port1 =
        TCP,
        "EtherNet/IP port",
        "20 F5 24 01",
        1;

```

Halaman ini sengaja dikosongkan

RIWAYAT PENULIS



Ivan Dharmawan Setianto, yang lahir di kota Malang pada tanggal 30 Juni 1997. Anak pertama dari dua bersaudara. mempunyai kampung halaman di Jambangan, Surabaya dan Pondok Belimbing Indah, Malang. Pernah menempuh Pendidikan di SDS Angkasa 9, SMPN 157 Jakarta Timur, dan SMAN 81 Jakarta Timur. Sekarang ini sedang menempuh jenjang Pendidikan pada bidang studi Komputer Kontrol di Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya. Menyukai kegiatan seperti, futsal, sepak bola dan jalan-jalan. Memiliki cita-cita menjadi orang yang berguna bagi semua orang dan sukses di dunia maupun akhirat.